

FP11

DIVIDE EXERCISER

MD-11-DCFPU-D

EP DCFPU D-DL-A

OCT 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

Made In U.S.A.

MACV-11-0000-0

FLOATING POINT DIVIDE EXERCISER

801

MACV11 27(732) 16-SEP-76 16:30 PAGE 1

REP 3

IDENTIFICATION

PRODUCT CODE: MAINDC-11-00FPJ-0-0
PRODUCT NAME: FP11 DIVIDE EXERCISER
DATE CREATED: 1-NOV-72
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: BEN CHAPMAN

COPYRIGHT ©, DIGITAL EQUIPMENT CORPORATION
1973

THIS MATERIAL IN THIS DOCUMENT IS FOR INFORMATION
PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT
SUPPORTED BY IT.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT.

CO1

MACY11 27(732) 16-SEP-76 16:30 PAGE 3

FLOATING POINT DIVIDE EXERCISER

MAINDEC-11-DCFFJ-0-0
TABLE OF CONTENTS

FPII DIVIDE EXERCISER

PAGE 2

CONTENTS

1. ABSTRACT
2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRELIMINARY PROGRAMS
3. LOADING PROCEDURE
4. STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS
 - 4.2 STARTING ADDRESS
 - 4.3 PROGRAM AND/OR OPERATOR ACTION
5. OPERATING PROCEDURE
 - 5.1 OPERATIONAL SWITCH SETTINGS
 - 5.2 SUBROUTINE ABSTRACT
6. ERRORS
7. RESTRICTIONS
8. MISCELLANEOUS
 - 8.1 EXECUTION TIME
 - 8.2 STACK POINTER
 - 8.3 POWER FAIL
9. PROGRAM DESCRIPTION

MAINDEC-11-DCFPD-D-0
DESCRIPTION

FP11 DIVIDE EXERCISER

PAGE 3

1. ABSTRACT

THIS PROGRAM EXERCISES THE FP11 FLOATING POINT DIVIDE INSTRUCTIONS (DIVF AND DIVD) WITH RANDOM NUMBERS. THE ANSWERS ARE CHECKED AGAINST RESULTS OBTAINED USING THE CORRESPONDING FORTRAN SOFTWARE ROUTINES.

2. REQUIREMENTS

2.1 EQUIPMENT

POP11 45 STANDARD COMPUTER WITH FP11 OPTION

2.2 STORAGE

THE ROUTINES USE MEMORY LOCATIONS C - 17500. THE END OF THE LISTINGS SHOWS THE ABSOLUTE LOCATIONS OF THE FORTRAN MATH ROUTINES WHICH WERE ASSEMBLED SEPARATELY AND LINKED TO THE MAIN PROGRAM VIA LINK11 ON A DEC SYSTEM.

2.3 PRELIMINARY PROGRAMS

DCFPD THRU DCFPL

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ASS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 CALL DOWN FOR WORST CASE TESTING

4.2 STARTING ADDRESS

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200.

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY USING ASS LOADER.
2. LOAD ADDRESS 200.

E01

MACY11 270732 16-SEP-76 16:30 PAGE 4

UNRECORDED-0000-0

FLIGHTING POINT DIVIDE EXERCISE

10

3) SET SWITCHES SEE S.1.1. ALL CORN FOR CORN CASE.
4) FEEDS 5-100.

MAINDEC-11-DCFPL-3-D FP11 DIVIDE EXERCISER
DESCRIPTION

PAGE 4

5) THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS.
6) THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT.
THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE
DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

SW 15	=	1		HLT ON ERROR
SW 14	=	1		SCOPE LOOP
SW 13	=	1		INHIBIT PRINTOUT
SW 12	=	1		INHIBIT TRACE TRAPPING
SW 11	=	1		INHIBIT ITERATIONS OF SUBTEST
SW 10	=	1		BELL ON ERROR
				BELL ON PASS COMPLETE
SW 09	=	0		CORE IMAGE TYPE-OUT 16 BIT WORDS
		0		FLOATING POINT TYPE-OUT (SIGN, EXPONENT, MANTISSA)
SW 08	=	1		LOOP ON TEST IN SW 7:0
		0		LOAD SW 7:0 INTO LS REGISTER

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE
TEST SECTION. IT RECORDS THE STARTING ADDRESS OF EACH
SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". WHEN
SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOADED
UPON. SW(11) ON A 1 INHIBITS ITERATION OF SUBTESTS.
CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST
SUCCESSFULLY COMPLETED. LAD IS UPDATED INSIDE EACH SUBTEST
AFTER THE FORTRAN ANSWER IS CALCULATED. SO THAT THE
ITERATIONS WILL INCLUDE ONLY THE FP11 PORTION OF THE TEST.

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE SEE 5.1.1. TO
INHIBIT TYPEOUTS, PUT SW 13 ON A 1.

5.2.3 TRTRAP

IF SW 12 IS ON A 0, THE T BIT WILL BE SET ON ALTERNATE
PASSES. WHEN SET, IT CAUSES A TRAP AFTER EACH INSTRUCTION.
THE FIRST INSTRUCTION EXECUTED UPON TRAPPING IS AN "RAC",
WHICH RETURNS TO THE INTERRUPTED SEQUENCE OF INSTRUCTIONS.
THIS SEQUENCE IS CONTINUED UNTIL THE END OF THE PROGRAM IS
REACHED.

000000-0000-0

FLOATING POINT DIVIDE EXERCISE

GO1

MACY11 27:732: 16-SEP-76 16:30 PAGE 6

100

REACHED.

MAINDEC-11-DCFPU-D-C
DESCRIPTION

FP11 DIVIDE EXERCISER

PAGE 5

5.2.4 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM J = 775 TO 999. ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR + 2.

5.2.5 FLOATING POINT TRAP (TC 244)

SINCE SOME OF THE SUBTESTS HAVE INTERRUPTS ENABLED, THE FLOATING POINT TRAP (F.TERR) CHECKS TO SEE IF FORTRAN ALSO GOT AN ERROR CONDITION. IF FORTRAN DIDN'T INDICATE AN ERROR, OR INTERRUPTS WERE DISABLED AN ERROR MIGHT OCCUR (SEE 5.2.2). IF AN INTERRUPT WAS ANTICIPATED, BUT DIDN'T OCCUR, THE SUBTEST WILL DETECT THE ERROR.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADDRESS, OPERAND, OPERATOR, OPERAND, EQUALS
FPP: ANSWER, FPS, FEC, FEA
FORTRAN: ANSWER, FPS, FEC, FEA

WHERE:

ADDRESS = ADDRESS OF ERROR HLT
OPERAND = RANDOM FLOATING POINT NUMBER INPUTS
OPERATOR = ARITHMETIC OPERATOR (+ OR -)
EQUALS = (=)
ANSWER = FLOATING POINT ANSWER
FPS = FLOATING POINT STATUS
FEC = FLOATING EXCEPTION CODES (ERROR CODES)
FEA = FLOATING EXCEPTION ADDRESS (ERROR ADDRESS)

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

MAINDEC-11-DCFPD-D-C
DESCRIPTION

FP11 DIVIDE EXERCISER

PAGE 6

8. MISCELLANECUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 15 SECONDS WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 600

8.3 POWER FAIL

EACH TEST CAN BE POWER FAILED WITH NO ERRORS EXCEPT ON THE FEC AND FEA. TO USE, START THE TEST AS USUAL AND POWER DOWN THEN UP AT ANY TIME. THE PROGRAM SHOULD TYPE "POWER" AND CONTINUE TO RUN WITH NO OTHER TYPECUTS.

9. PROGRAM DESCRIPTION

THIS PROGRAM TESTS THE DIVIDE INSTRUCTIONS ON THE FP11 IN ROUND AND TRUNCATE MODES AND WITH INTERRUPTS ON AND OFF. EACH PROGRAM HAS MANY SUBTESTS (THE CODE BETWEEN 2 SCOPE STATEMENTS) WHICH ARE RUN 256 TIMES BEFORE CONTINUING TO THE NEXT. SW(11) ON A 1 CAUSES EACH SUBTEST TO BE RUN ONLY ONCE. THE ADDRESS ICNT (LOC 1000) AND DISPLAY REGISTER ON THE 11/45 EACH CONTAIN THE ITERATION COUNT IN THE LEFT BYTE AND THE TEST NUMBER IN THE RIGHT BYTE. ALL THE SUBTESTS SHOULD BE RUN SEQUENTIALLY BY STARTING AT 200 NOT BY STARTING AT THE BEGINNING OF THE SUBTEST. TO LOOP ON A PARTICULAR SUBTEST, PUT THE TEST NUMBER (SEE LISTING) IN THE RIGHT BYTE OF THE SWITCH REGISTER AND SW(8) ON A 1. THIS TEST WILL BE LOOPED UPON UNTIL SW(8) IS PUT ON A 0 OR THE RIGHT BYTE IS CHANGED. IF THE TEST IS NON-EXISTANT, THE PROGRAM WILL BE RUN AS USUAL.

THE FORTRAN MATH ROUTINES WERE TAKEN UNMODIFIED FROM THE PDP-11 FORTRAN PACKAGE AND ASSEMBLED AS SEPERATE MODULES. THEY WERE LINKED TO THE MAIN PROGRAMS VIA LNKX11 ON A DECSYSTEM-10 WHICH PRODUCES A BINARY TAPE IN THE NORMAL ABSOLUTE FORMAT. THUS, THE PROGRAMS LOAD AND RUN JUST LIKE ANY OTHER DIAGNOSTIC PROGRAM.

NOTE: SINCE THE FP11 LOAD, STORE AND COMPARE INSTRUCTIONS (LOF, LDD, STF, STD, CMPE AND CMPC) ARE ALSO USED IN THIS PROGRAM, IT IS POSSIBLE THAT THEY AND NOT THE DIVIDE INSTRUCTIONS COULD CAUSE ERRORS.

.ENDR

000000

.TITLE MAINDEC-11-DCFPD-D FLOATING POINT DIVIDE EXERCISER
.ASECT
.GLOBL \$DVR,\$DVD,\$ERRA
:COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
:PROGRAM BY KEN CHAPMAN

SWITCH	USE
8	0 - LOAD UB REGISTER WITH SW(7:0)
9	1 - LOOP ON TEST IN SW(7:0)
10	TTY OUTPUT FORMAT: 0 - SIGN, EXPONENT, MANTISSA 1 - CORE IMAGE (16 BIT WORDS)
11	0 - BELL ON PASS COMPLETE 1 - BELL ON ERROR
12	INHIBIT ITERATIONS
13	INHIBIT TRACE TRAP
14	INHIBIT ERROR TYPEOUTS
15	LOOP ON TEST HALT ON ERROR

:OUTPLT FORM:

: ADDRESS, OPERAND, OPERATOR, OPERAND, EQUALS
: FPP: ANSWER, FPS, FEC, FEA
: FORTRAN:

BIT	FPS	REASON	CODE	FEC	ERROR
0		CARRY	0		ADDRESS ERROR
1		OVERFLOW	2		OPCODE ERROR
2		ZERO	4		DIVIDE BY ZERO
3		NEGATIVE	6		CONVERSION ERROR
4		MAINTAINANCE MODE	10		OVERFLOW
5		TRUNCATE MODE	12		UNDERFLOW
6		LONG INTEGER MODE	14		UNDEFINED VARIABLE
7		DOUBLE PRECISION MODE	16		UBREAK TRAP
8		INTERUPT ON CONVERSION ERROR			
9		INTERUPT ON OVERFLOW			
10		INTERUPT ON UNDERFLOW			
11		INTERUPT ON UNDEFINED VARIABLE			
12					
13					
14		INTERUPT ENABLE			
15		ERROR FLAG			

001022	000000			ANS1:	DUMMY	
001024	000000				DUMMY	
001026	000000				DUMMY	
001030	000000				DUMMY	
001032	000000			ANS2:	DUMMY	
001034	000000				DUMMY	
001036	000000				DUMMY	
001040	000000				DUMMY	
001042	000000			FPS:	0	:FLOATING POINT STATUS
001044	000000			FEC:	0	:FLOATING EXCEPTION CODES
001046	000000			FPC:	0	:FLOATING PC
001050	000000			SFPS:	0	:FORTRAN FLOATING POINT STATUS
001052	000000			SFEC:	0	:FORTRAN FLOATING EXCEPTION CODES
001054	000000			SFPC:	0	:FORTRAN FLOATING PC
001056	012706	000600		BEG:	MOV	#600,SP
001062	012737	001104	000004		MOV	#M112C,2#4
001070	005737	177772			TST	2#177772
001074	012767	000006	005034		MOV	#6,YESRT
001102	000403				BR	BEGIN
001104	016737	010050	000010	M112C:	MOV	FPTADR,2#10
						:LOAD THE ILLEGAL INSTRUCTION VECTOR
						: WITH THE ADDRESS OF THE FPU.
						: THE FPU WILL HANDLE THE BAD OPCODES
						:RESET 4
001112	012737	000006	000004	BEGIN:	MOV	#6,2#4
001120	012706	000600			MOV	#600,SP
001124	012737	006136	000014		MOV	#YESRT,2#14
001132	012777	010606	010026		MOV	#POWDWN,3DOWNVEC
001140	012777	000340	010022		MOV	#340,DOWNVEC+2
001146	012737	011006	000020		MOV	#101,2#20
001154	012700	000030			MOV	#30,RO
001160	012720	007542			MOV	#TRAP,(0)+
001164	012720	000340			MOV	#340,(0)+
001170	012720	006140			MOV	#EMT,(0)+
001174	012710	000340			MOV	#340,(0)
001200	012777	007210	007754		MOV	#FLTERF,OFFVECT
001206	012777	000340	007750		MOV	#340,OFFVECT+2
001214	005067	177560			CLR	ICNT
001220	005067	007756			CLR	LAD
						:SET TRACE TRAP VECTOR
						:SET UP VECTOR 20
						:SET RO TO VECTOR 30
						:SET EMT VECTOR
						:SET TRAP VECTOR
						:LOAD INTERRUPT VECTOR
						:LOCK UP PROCESSOR

```

*****
:TEST 1:      EXERCISE DIVF (DIVIDE FLOATING)
:              ALL INTERRUPTS ON
:              ROUNDING MODE
*****

```

001224	104400			SCOPE		
001226	012767	007400	177614	MOV	#007400,\$FPS	;SET IE BITS IN FORTRAN ANSWER
001234	005067	177612		CLR	\$FEC	;CLR FORTRAN FEC
001240	005067	177610		CLR	\$FPC	;CLR FORTRAN FPC
001244	005067	177572		CLR	FPS	;CLR FPU FPS BUFFER
001250	005067	177570		CLR	FEC	;CLR FPU FEC BUFFER
001254	005067	177566		CLR	FPC	;CLR FPU FPC BUFFER
001260	004767	005762		JSR	PC, RANDM2	;GET RANDOM INPUT DATA
001264	004467	005010		JSR	R4, \$POLSH	;ENTER POLISH MODE
001270	006312			\$P.2A		;PUSH 2 WORDS ON STACK (LONUM)
001272	006334			\$P.2B		;PUSH 2 WORDS ON STACK (HINUM)
001274	000000G			\$DVR		;ADDRESS OF FORTRAN DIVIDE
001276	006402			\$TST		;DETERMINE THE CONDITION CODES
001300	006346			\$POP2X		;POP 2 WORDS AND EXIT POLISH MODE
001302	016700	177542		MOV	\$FPS, RC	;DISPLAY FLOATING POINT STATUS
001306	170127	040000		LDFPS	#040000	;SET INTERRUPT DISABLE
001312	172467	177464		LDF	LONUM, ACO	;LOAD ACO WITH A RANDOM NUMBER
001316	172567	177470		LDF	HINUM, AC1	;LOAD AC1 WITH A RANDOM NUMBER
001322	172767	177504		LDF	ANS2, AC3	;LOAD AC3 WITH THE SUM
001326	170127	007400		LDFPS	#007400	;TURN INTERRUPTS ON
001332	012767	001340	007642	MOV	*,+6, LAD	;RESET LOOP ADDRESS

001340	172600			LDF	ACO, AC2	;LOAD ACO INTO AC2
001342	174601			DIVF	AC1, AC2	;DIVIDE AC1 INTO AC2
001344	005767	177500		TST	\$FPS	;CHECK FOR FORTRAN FPS ERROR FLAG
001350	100412			BMI	ERR1	;BRANCH IF ERROR FLAG SET
001352	170267	177464		STFPS	FPS	;STORE FLOATING POINT STATUS
001356	026767	177460	177464	CMP	FPS, \$FPS	;CHECK FPS
001364	001427			BEQ	TST1	;BRANCH IF OK
001366	174267	177430		STF	AC2, ANS1	;SAVE FPU ANSWER
001372	104001			HLT+1		;FPS ERROR
001374	000444			BR	END1	;SKIP COMPARE
001376	170000			CFCC		;WAIT FOR FPU TO FINISH
001400	026767	177436	177442	CMP	FPS,\$FPS	;CHECK THE FLOATING POINT STATUS
001406	001402			BEQ	*,+6	;BRANCH IF OK
001410	104377			HLT+377		;FPS ERROR
001412	000435			BR	END1	;SKIP TO END
001414	026767	177424	177430	CMP	FEC,\$FEC	;CHECK THE FLOATING EXCEPTION CODES
001422	001402			BEQ	*,+6	;BRANCH IF OK
001424	104377			HLT+377		;FEC IS WRONG
001426	000427			BR	END1	;SKIP TO END
001430	026767	177412	177416	CMP	FPC,\$FPC	;CHECK FLOATING PC
001436	001402			BEQ	TST1	;BRANCH IF OK

MACY11-20FPL-3
MACY11-20FPL-3FLOATING POINT DIVIDE EXERCISER
TEST SECTION

```

001440 104377 HLT+377 ;WRONG ADDRESS IN FPC
001442 000421 BR END1 ;SKIP TO END

001444 173702 TST1: CMPF AC2, AC3 ;COMPARE FPU ANSWER TO FORTRAN ANSWER
001446 170000 CFCC ;COPY FLOATING CONDITION CODES
001450 001416 BEQ END1 ;ANSWERS CHECK
;COMPENSATE FOR FORTRAN INACCURACIES.
001452 174267 177344 STF AC2, ANS1 ;SAVE FPU ANSWER
001456 162767 000001 177340 SUB #1, ANS1+2 ;DECREMENT FPU ANSWER
001464 005667 177332 SBC ANS1, AC3 ;CHECK ANSWERS AGAIN
001470 173767 177326 CMPF ANS1, AC3 ;COPY FLOATING CONDITION CODES
001474 170000 CFCC ;BRANCH IF OK
001476 001403 BEQ END1 ;SAVE FPU ANSWER
001500 174267 177316 STF AC2, ANS1 ;FPU AND FORTRAN DISAGREE
001504 104000 HLT

001506 005067 177330 END1: CLR FPS ;CLR FPU FPS BUFFER
001512 104400 SCOPE

```

```

*****
:TEST 2: EXERCISE DIVD (DIVIDE DOUBLE PRECISION)
: ALL INTERRUPTS ON
: ROUNDING MODE
*****

```

```

001514 012767 007600 177326 MOV #007600,$FPS ;SET IE BITS IN FORTRAN ANSWER
001522 005067 177324 CLR $FEC ;CLR FORTRAN FEC
001526 005067 177322 CLR $FPC ;CLR FORTRAN FPC
001532 005067 177304 CLR FPS ;CLR FPU FPS BUFFER
001536 005067 177302 CLR FEC ;CLR FPU FEC BUFFER
001542 005067 177300 CLR FPC ;CLR FPU FPC BUFFER
001546 004767 005632 JSR PC, RANDM4 ;GET RANDOM INPUT DATA
001552 004467 004522 JSR R4, $POLSH ;ENTER POLISH MODE
001556 006302 $P.4A ;PUSH 4 WORDS ON STACK (LONUM)
001560 006324 $P.4B ;PUSH 4 WORDS ON STACK (HINUM)
001562 000000 $DIVD ;ADDRESS OF FORTRAN DIVIDE
001564 006402 $TST ;DETERMINE THE CONDITION CODES
001566 006360 $POP4X ;POP 4 WORDS AND EXIT POLISH MODE

001570 016700 177254 MOV $FPS, R0 ;DISPLAY FLOATING POINT STATUS
001574 170127 040200 LDFPS #040200 ;SET FID AND FD
001600 172467 177176 LDD LONUM, AC0 ;LOAD AC0 WITH A RANDOM NUMBER
001604 172567 177202 LDD HINUM, AC1 ;LOAD AC1 WITH A RANDOM NUMBER
001610 172767 177216 LDD ANS2, AC3 ;LOAD AC3 WITH THE SUM
001614 170127 007600 LDFPS #007600 ;TURN INTERRUPTS ON
001620 012767 001626 MOV #.46, LAD ;RESET LOOP ADDRESS

```

```

001626 172600 FET2: LDD AC0, AC2 ;LOAD AC0 INTO AC2
001630 174600 DIVD AC1, AC2 ;DIVIDE AC1 INTO AC2
001634 005067 $FPS ;CHECK FOR FORTRAN FPS ERROR FLAG
001638 177316 BMI ERR2 ;BRANCH IF ERROR FLAG SET
001642 177316 STFPS FPS ;STORE FLOATING POINT STATUS
001646 177316 CMP FPS, $FPS ;CHECK FPS
001650 177316 BEQ TST2 ;BRANCH IF OK

```


FLOATING POINT DIVIDE EXERCISER TEST SECTION

001654	174267	177142	STD	AC2.	ANS1	:SAVE FPU ANSWER	
001655	174267		HLT+1			:FPS ERROR	
001656	000450		BR	END2		:SKIP TO END	
001664	170000	177150	ERR2:	CFEC		:WAIT FOR FPU TO FINISH	
001665	026767	177154	CMP	FPS,SFPS		:CHECK THE FLOATING POINT STATUS	
001666	001402		BEG	+6		:BRANCH IF OK	
001667	104377		HLT+377			:FPS ERROR	
001668	000441		BR	END2		:SKIP TO END	
001702	026767	177136	CMP	FEC,SFEC		:CHECK THE FLOATING EXCEPTION CODES	
001703	001402		BEG	+6		:BRANCH IF OK	
001704	104377		HLT+377			:FEC IS WRONG	
001705	000433		BR	END2		:SKIP TO END	
001716	026767	177124	CMP	FPC,SFPC		:CHECK FLOATING PC	
001717	001402		BEG	+5		:BRANCH IF OK	
001718	104377		HLT+377			:WRONG ADDRESS IN FPC	
001719	000425		BR	END2		:SKIP TO END	
001732	173702	177124	TEST2:	CMPO	AC2.	AC3	:COMPARE FPU ANSWER TO FORTRAN ANSWER
001733	170000		CFEC			:COPY FLOATING CONDITION CODES	
001734	001422		BEG	END2		:ANSWERS CHECK	
001740	174267	177056	STD	AC2.	ANS1	:SAVE FPU ANSWER	
001741	162767	000001	HLT			:DECREMENT FPU ANSWER	
001742	005667	177050	SBC	ANS1+4			
001743	005667	177042	SBC	ANS1+2			
001744	005667	177034	SBC	ANS1			
001745	173767	177030	CMPO	ANS1.	AC3	:CHECK ANSWERS AGAIN	
001746	170000		CFEC			:COPY FLOATING CONDITION CODES	
001747	001403		BEG	END2		:BRANCH IF OK	
001748	174267	177020	STD	AC2.	ANS1	:SAVE FPU ANSWER	
001749	104300		HLT			:FPU AND FORTRAN DISAGREE	
001750	005067	177032	END2:	CLR	FPS	:CLR FPU FPS BUFFER	
001751	001403		SCOPE				

:TEST 3: EXERCISE DIVE (DIVIDE FLOATING)
: OVERFLOW AND UNDERFLOW INTERRUPTS OFF.
: ROUNDING MODE
:*****

002012	012767	004400	177030	MOV	8004400,SFPS	:SET IE BITS IN FORTRAN ANSWER
002013	005067	177026		CLR	SFEC	:CLR FORTRAN FEC
002014	005067	177024		CLR	SFPC	:CLR FORTRAN FPC
002015	005067	177006		CLR	FPS	:CLR FPU FPS BUFFER
002016	005067	177004		CLR	FEC	:CLR FPU FEC BUFFER
002017	005067	177002		CLR	FPC	:CLR FPU FPC BUFFER
002018	004767	005176		CLR	PC	:GET RANDOM INPUT DATA
002019	004767	004224		CLR	PC	:ENTER POLISH MODE
002020	006012			STD	20	:PUSH 2 WORDS ON STACK
002021	006012			STD	20	:PUSH 2 WORDS ON STACK
002022	006012			STD	20	:ADDRESS OF FORTRAN ANSWER
002023	006012			STD	20	:CHECK THE CONDITION CODES

MACY11 27(732) 16-SEP-76
PAGE 15

FLOATING POINT DIVIDE EXERCISE
FPU SECTION

002064	006346			SPOP2X		:POP 2 WORDS AND EXIT POLISH MODE
002066	016700	176756		MOV	SFPS	PC
002068	174267	176756		LOFPS	8040000	:DISPLAY FLOATING POINT STATUS
002070	174267	176756		LOF	LONUM.	AC0
002072	174267	176756		LOF	HINUM.	AC1
002074	174267	176756		LOF	ANS2.	AC3
002076	174267	176756		LOFPS	8004400	:TURN INTERRUPTS ON EXCEPT OVERFLOW AND UNDERFLOW
002078	174267	176756		MOV	8.+6.	LAD
002080	174267	176756				:RESET LOOP ADDRESS

002102	172600			LOF	AC0.	AC2	:LOAD AC0 INTO AC2
002104	174267			LOF	AC1.	AC2	:DIVIDE AC1 INTO AC2
002106	005767	176714		LOF	SFPS		:CHECK FOR FORTRAN FPS ERROR FLAG
002108	100412			LOF	ERR3		:BRANCH IF ERROR FLAG SET
002110	170267	176700		LOF	SFPS		:STORE FLOATING POINT STATUS
002112	026767	176674	176700	LOF	SFPS		:CHECK FPS
002114	001427			LOF	IS3		:BRANCH IF OK
002116	174267	176644		LOF	AC2.	ANS1	:SAVE FPU ANSWER
002118	104001			LOF	+		:FPS ERROR
002120	000450			BR	ENC3		:SKIP COMPARE
002162	170000			CFEC			:WAIT FOR FPU TO FINISH
002164	026767	176652	176656	CFEC	FPS,SFPS		:CHECK THE FLOATING POINT STATUS
002166	001403			BEQ	+.6		:BRANCH IF OK
002168	104377			HLT	+377		:FPS ERROR
002170	000441			BR	ENC3		:SKIP TO END
002200	026767	176640	176644	CFEC	FEC,SFEC		:CHECK THE FLOATING EXCEPTION CODES
002202	001403			BEQ	+.6		:BRANCH IF OK
002204	104377			HLT	+377		:FEC IS WRONG
002206	000433			BR	ENC3		:SKIP TO END
002214	026767	176626	176632	CFEC	FPC,SFPC		:CHECK FLOATING PC
002216	001403			BEQ	TST3		:BRANCH IF OK
002218	104377			HLT	+377		:WRONG ADDRESS IN FPC
002220	000425			BR	ENC3		:SKIP TO END
002230	032767	000002	176612	BIT	82	SFPS	:CHECK FOR OVERFLOW
002232	001021			BNE	ENC3		:BRANCH IF OVERFLOW
002234	173702			CMPE	AC2.	AC3	:COMPARE FPU ANSWER TO FORTRAN ANSWER
002236	170000			CFEC			:COPY FLOATING CONDITION CODES
002238	001416			BEQ	ENC3		:ANSWERS CHECK
002240							:COMPENSATE FOR FORTRAN INACCURACIES.
002242	174267	176550		LOF	AC2.	ANS1	:SAVE FPU ANSWER
002244	162767	176550	176554	LOF	81	ANS1+2	:DECREMENT FPU ANSWER
002246	005667	176536		LOF	ANS1		
002248	173767	176532		LOF	ANS1.	AC3	:CHECK ANSWERS AGAIN
002250	170000			CFEC			:COPY FLOATING CONDITION CODES
002252	001403			BEQ	ENC3		:BRANCH IF OK
002254	174267	176522		LOF	AC2.	ANS1	:SAVE FPU ANSWER
002256	104001			HLT			:FPU AND FORTRAN DISAGREE
002302	005067	176504		ENC3:	CLR	FPS	:CLR FPU FPS BUFFER
002304	004000				SCOPE		

MACY11 27(732) 16-SEP-76 16:30 PAGE 16

MACY11 27(732) 16-SEP-76 16:30 PAGE 16

002

MACY11 27(732) 16-SEP-76 16:30 PAGE 16

```

*****
TEST 4:      EXERCISE DIVD (DIVIDE DOUBLE PRECISION)
              OVERFLOW AND UNDERFLOW INTERRUPTS OFF
              ROUNDING MODE
*****

002331 012767 004600 176532      MOV      #004600, $FPS      :SET IE BITS IN FORTRAN ANSWER
002332 005067 176530      CLR      $FEC          :CLR FORTRAN FEC
002333 005067 176526      CLR      $FPC          :CLR FORTRAN FPC
002334 005067 176510      CLR      $FPS          :CLR FPU FPS BUFFER
002335 005067 176506      CLR      $FEC          :CLR FPU FEC BUFFER
002336 005067 176504      CLR      $FPC          :CLR FPU FPC BUFFER
002337 004767 005036      GET      $RNDM          :GET RANDOM INPUT DATA
002338 004467 003726      ENTER     $POLISH        :ENTER POLISH MODE
002339 006302      PUSH     4($LNUM)              :PUSH 4 WORDS ON STACK ($LNUM)
002340 006324      PUSH     4($MINUM)             :PUSH 4 WORDS ON STACK ($MINUM)
002341 000000      ADDRESS   $DIVD                :ADDRESS OF FORTRAN DIVIDE
002342 006402      DETERMINE $COND               :DETERMINE THE CONDITION CODES
002343 006360      POP      4($LNUM)              :POP 4 WORDS AND EXIT POLISH MODE

002364 016700 176460      MOV      $FPS, RC      :DISPLAY FLOATING POINT STATUS
002365 170127 040200      LDFPS   #040200        :SET FID AND FD
002366 172467 176402      LODD     $LNUM, ACC     :LOAD ACC WITH A RANDOM NUMBER
002367 172567 176406      LODD     $MINUM, AC1    :LOAD AC1 WITH A RANDOM NUMBER
002368 172767 176422      LODD     $ANS2, AC3     :LOAD AC3 WITH THE SUM
002369 170127 004600      LDFPS   #004600        :TURN INTERRUPTS ON, EXCEPT OVER AND UNDERFLOW
002370 012767 002422      MOV      #. +6, LAC     :RESET LOOP ADDRESS

*****

002422 176600      FE*4:  DIVD     ACC, AC2      :LOAD ACC INTO AC2
002423 174601      DIVD     AC1, AC2          :DIVIDE AC1 INTO AC2
002424 005767 176416      LDFPS   $FPS, $FEC    :CHECK FOR FORTRAN FPS ERROR FLAG
002425 100412      BR*4     $ERR4             :BRANCH IF ERROR FLAG SET
002426 170267 176402      LDFPS   $FPS, $FPS     :STORE FLOATING POINT STATUS
002427 176376 176402      LDFPS   $FPS, $FPS     :CHECK FPS
002428 001427      BR*4     $ERR4             :BRANCH IF OK
002429 174267 176346      LDFPS   $ANS1, $ANS1   :SAVE FPU ANSWER
002430 104001      LDFPS   $FPS, $FPS     :FPS ERROR
002431 000454      BR*4     $ERR4             :SKIP COMPARE

002460 170000      ERR4:  CFEC     FPS, $FEC     :WAIT FOR FPU TO FINISH
002461 026767 176354 176360      CMP     .+6, $FEC :CHECK THE FLOATING POINT STATUS
002462 001402      BEQ     $ERR4             :BRANCH IF OK
002463 104377      HL*+377      :FPS ERROR
002464 000445      BR*4     $ERR4             :SKIP TO END

002476 026767 176342 176346      CMP     $FEC, $FEC :CHECK THE FLOATING EXCEPTION CODES
002477 001402      BEQ     .+6              :BRANCH IF OK
002478 104377      HL*+377      :FEC IS WRONG
002479 000437      BR*4     $ERR4             :SKIP TO END

002480 026767 176330 176334      CMP     $FEC, $FEC :CHECK FLOATING PC
002481 001402      BEQ     $ERR4             :BRANCH IF OK
002482 104377      HL*+377      :WRONG ADDRESS IN FEC

```

MACY11-27(732)-0
16-SEP-76

FLOATING POINT DIVIDE EXERCISER TEST SECTION

E02

MACY11 27(732) 16-SEP-76 16:30 PAGE 17

```

002524 000431 BR EN04 :SKIP TO END
002526 032767 000002 176314 TST4: BIT B2 $FPS :CHECK FOR OVERFLOW
002528 001026 BNE EN04 :BRANCH IF OVERFLOW
002530 173732 CMPO AC2, AC3 :COMPARE FPU ANSWER TO FORTRAN ANSWER
002532 170000 CFCC :COPY FLOATING CONDITION CODES
002534 001422 BEG EN04 :ANSWERS CHECK
:COMPENSATE FOR FORTRAN INACCURACIES.
002536 174267 176252 STD AC2, ANS1 :SAVE FPU ANSWER
002538 163767 000001 176252 SUB B1 ANS1+6 :DECREMENT FPU ANSWER
002540 005667 176244 SBC ANS1+4
002542 005667 176236 SBC ANS1+2
002544 005667 176230 SBC ANS1
002546 173767 176224 CMPO ANS1, AC3 :CHECK ANSWERS AGAIN
002548 170000 CFCC :COPY FLOATING CONDITION CODES
002550 001403 EN04 :BRANCH IF OK
002552 174267 176214 SUB AC2, ANS1 :SAVE FPU ANSWER
002554 104000 HLT :FPU AND FORTRAN DISAGREE
002556 005067 176226 EN04: CLR FPS :CLR FPU FPS BUFFER
002558 104400 SCOPE

```

TEST 5: EXERCISE DIVF (DIVIDE FLOATING)
ALL INTERRUPTS ON
TRUNCATE MODE

```

002616 012767 007440 176224 MOV #007440, $FPS :SET IE BITS IN FORTRAN ANSWER
002618 005067 176220 WFE0 :CLR FORTRAN FEC
002620 005067 176220 WFP0 :CLR FORTRAN FPC
002622 005067 176220 FPS :CLR FPU FPS BUFFER
002624 005067 176220 FEC :CLR FPU FEC BUFFER
002626 005067 176176 FPC :CLR FPU FPC BUFFER
002628 004767 004372 YSR PC, RAN0M2 :GET RANDOM INPUT DATA
002630 004467 003420 YSR R4, $POLSH :ENTER POLISH MODE
002632 006312 SP, 2B :PUSH 2 WORDS ON STACK (LONJM)
002634 006334 SP, 2B :PUSH 2 WORDS ON STACK (HINJM)
002636 000000G SOVR :ADDRESS OF FORTRAN DIVIDE
002638 006402 BTST :DETERMINE THE CONDITION CODES
002640 006346 SPOP2X :POP 2 WORDS AND EXIT POLISH MODE
002672 016700 176152 MOV $FPS, PC :DISPLAY FLOATING POINT STATUS
002674 170127 040000 LOFPS #040000 :SET FID
002676 172467 176074 LCF LONJM, ACC :LOAD ACC WITH A RANDOM NUMBER
002678 172567 176100 LCF HINJM, AC1 :LOAD AC1 WITH A RANDOM NUMBER
002680 172767 176114 LCF ANS2, AC3 :LOAD AC3 WITH THE SUM
002682 170127 007440 LOFPS #007440 :TURN INTERRUPTS ON
002684 012767 002730 MOV B, +6, LAD :RESET LOOP ADDRESS
002686 006252

```

```

002730 172600 176110 RETS: LCF ACC, AC2 :LOAD ACC INTO AC2
002732 174500 DIVF AC1, AC2 :DIVIDE AC1 INTO AC2
002734 003767 YST $FPS :CHECK FOR FORTRAN FPS ERROR FLAG
002736 100412 BMI ERPS :BRANCH IF ERROR FLAG SET

```

F02

MACY11-00FPU-0
TEST SECTION

FLOATING POINT DIVIDE EXERCISE
TEST SECTION

MACY11 27(732) 16-SEP-76 16:30 PAGE 19

002742	170267	176274		STFPS	FPS		:STORE FLOATING POINT STATUS
002744	026767	176070	176074	CMP	FPS	\$FPS	:CHECK FPS
002746	001427			BEG	TST5		:BRANCH IF OK
002748	174267	176040		STF	AC2.	ANS1	:SAVE FPU ANSWER
002750	104001			HLT+1			:FPS ERROR
002764	000455			BR	ENDS		:SKIP COMPARE
002766	170000						
002770	026767	176046	176052	ERRS: CFCC	FPS,\$FPS		:WAIT FOR FPU TO FINISH
002772	001402			CMP	..+6		:CHECK THE FLOATING POINT STATUS
002774	104377			BEG			:BRANCH IF OK
003000	000446			HLT+377			:FPS ERROR
003002				BR	ENDS		:SKIP TO END
003004	026767	176034	176040	CMP	FEC,\$FEC		:CHECK THE FLOATING EXCEPTION CODES
003012	001402			BEG	..+6		:BRANCH IF OK
003014	104377			HLT+377			:FEC IS WRONG
003016	000440			BR	ENDS		:SKIP TO END
003020	026767	176022	176026	CMP	FPC,\$FPC		:CHECK FLOATING PC
003026	001402			BEG	TST5		:BRANCH IF OK
003030	104377			HLT+377			:WRONG ADDRESS IN FPC
003032	000432			BR	ENDS		:SKIP TO END
003034	173702			TST5: CMPE	AC2.	AC3	:COMPARE FPU ANSWER TO FORTRAN ANSWER
003036	170000			CFCC			:COPY FLOATING CONDITION CODES
003040	001427			BEG	ENDS		:ANSWERS CHECK
003042	174267	175754		:COMPENSATE FOR FORTRAN			:INACCURACIES.
003046	062767	000001	175750	STF	AC2.	ANS1	:SAVE FPU ANSWER
003054	005567	175742		ADD	#1	ANS1+2	:INCREMENT FPU ANSWER
003060	173767	175736		ACC	ANS1.	AC3	:CHECK ANSWERS AGAIN
003064	170000			CMPE			:COPY FLOATING CONDITION CODES
003066	001414			CFCC			:BRANCH IF OK
003070	162767	000002	175726	BEG	ENDS		:DECREMENT FPU ANSWER
003076	005667	175720		SUB	#2	ANS1+2	
003102	173767	175714		SBC	ANS1		:CHECK ANSWERS AGAIN
003106	170000			CMPE	ANS1.	AC3	:COPY FLOATING CONDITION CODES
003110	001403			CFCC			:BRANCH IF OK
003112	174267	175704		BEG	ENDS		:SAVE FPU ANSWER
003116	104000			STF	AC2.	ANS1	:FPU AND FORTRAN DISAGREE
003120	005067	175716		HLT			
003124	104000			ENDS: CLR	FPS		:CLR FPU FPS BUFFER
				SCOPE			

TEST 6: EXERCISE DIVD (DIVIDE DOUBLE PRECISION)
ALL INTERRUPTS ON
TRUNCATE MODE

003126	012767	007640	175714	MOV	#007640,\$FPS		:SET IE BITS IN FORTRAN ANSWER
003130	005067	175712		CLR	\$FEC		:CLR FORTRAN FEC
003134	005067	175710		CLR	\$FPC		:CLR FORTRAN FPC
003138	005067	175672		CLR	FPS		:CLR FPU FPS BUFFER
003142	005067	175670		CLR	FEC		:CLR FPU FEC BUFFER
003146	005067	175668		CLR	FPC		:CLR FPU FPC BUFFER

MACY11 27(732) 16-SEP-76
 MACY11 27(732) 16-SEP-76

FLOATING POINT DIVIDE EXERCISE TEST SECTION

003160	004767	004220	JSR	PC.	RANDOM4	:GET RANDOM INPUT DATA
003164	004467	003110	JSR	PC.	SPOLSH	:ENTER POLISH MODE
003168	006302		SP.4A			:PUSH 4 WORDS ON STACK (LONUM)
003172	006324		SP.4E			:PUSH 4 WORDS ON STACK (MINUM)
003174	000000		SDVD			:ADDRESS OF FORTRAN DIVIDE
003176	006402		STST			:DETERMINE THE CONDITION CODES
003200	006360		SPOP4X			:POP 4 WORDS AND EXIT POLISH MODE

003202	016700	175642	MOV	SFPS.	RC	:DISPLAY FLOATING POINT STATUS
003206	170127	040200	LDFPS	8040200		:SET FID AND FC
003210	172467	175564	LDD	LONUM.	ACC	:LOAD ACC WITH A RANDOM NUMBER
003214	172567	175570	LDD	MINUM.	AC1	:LOAD AC1 WITH A RANDOM NUMBER
003218	172767	175604	LDD	ANS2.	AC3	:LOAD AC3 WITH THE SUM
003222	170127	007640	LDFPS	8007640		:TURN INTERRUPTS ON
003226	012767	003240	MOV	8.45.	LAC	:RESET LOOP ADDRESS

003240	172600		LDD	ACC.	AC2	:LOAD ACC INTO AC2
003242	174601		DIVC	AC1	AC2	:DIVIDE AC1 INTO AC2
003244	005767	175600	TEST	SFPS		:CHECK FOR FORTRAN FPS ERROR FLAG
003250	100412		BMI	ERR6		:BRANCH IF ERROR FLAG SET
003252	170267	175564	STFPS	FPS		:STORE FLOATING POINT STATUS
003256	026767	175560	CMP	FPS	SFPS	:CHECK FPS
003264	001427		BEG	5.6		:BRANCH IF OK
003266	174267	175530	STAC	AC2.	ANS1	:SAVE FPU ANSWER
003272	104001		HLT+1			:FPS ERROR
003274	000465		BR	END6		:SKIP TO END

003276	170000		CFCC			:WAIT FOR FPU TO FINISH
003300	026767	175536	CMP	FPS, SFPS		:CHECK THE FLOATING POINT STATUS
003306	001402		BEG	5.6		:BRANCH IF OK
003310	104377		HLT+377			:FPS ERROR
003312	000456		BR	END6		:SKIP TO END

003314	026767	175524	CMP	FEC, SFEC		:CHECK THE FLOATING EXCEPTION CODES
003322	001402		BEG	5.6		:BRANCH IF OK
003324	104377		HLT+377			:FEC IS WRONG
003326	000450		BR	END6		:SKIP TO END

003330	026767	175512	CMP	FPC, SFPC		:CHECK FLOATING PC
003336	001402		BEG	5.6		:BRANCH IF OK
003340	104377		HLT+377			:WRONG ADDRESS IN FPC
003342	000442		BR	END6		:SKIP TO END

003344	173702		TEST:	CMPD	AC2.	AC3	:COMPARE FPU ANSWER TO FORTRAN ANSWER
003346	170000			CFCC			:COPY FLOATING CONDITION CODES
003350	001437			BEG	END6		:ANSWERS CHECK

003352	174267	175444	STAC	AC2.	ANS1		:SAVE FPU ANSWER
003356	062767	000001	ADD	81	ANS1+6		:INCREMENT FPU ANSWER
003364	005567	175436	ADD	ANS1+4			
003370	005567	175430	ADD	ANS1+2			
003374	005567	175422	ADD	ANS1			
003378	173702	175416	CMPD	ANS1.	AC3		:CHECK ANSWERS AGAIN
003380	170000		CFCC				:COPY FLOATING CONDITION CODES

H02

MACY11 27(732) 16-SEP-76 16:30 PAGE 20

MACY11 27(732) 16-SEP-76

FLOATING POINT DIVIDE EXERCISER
TEST SECTION

003426	001420		BEG	END6		:BRANCH IF OK
003427	162767	003000	SUB	#2	ANS1+5	:DECREMENT FPU ANSWER
003428	005667	175412	SUB	ANS1+4		
003429	005667	175404	SUB	ANS1+2		
003430	005667	175376	SUB	ANS1		
003431	005667	175376	SUB	ANS1		
003432	173767	175364	CHPD	ANS1.	AC3	:CHECK ANSWERS AGAIN
003433	170000		CFCC			:COPY FLOATING CONDITION CODES
003434	001420		BEG	END6		:BRANCH IF OK
003435	174267	175354	STD	AC2.	ANS1	:SAVE FPU ANSWER
003436	104000		HLT			:FPU AND FORTRAN DISAGREE
003437	005067	175366	END6:	CLR	FPS	:CLR FPU FPS BUFFER
003438	104000		SCOPE			

 TEST 7: EXERCISE DIVF (DIVIDE FLOATING)
 OVERFLOW AND UNDERFLOW INTERRUPTS OFF.
 TRUNCATE MODE

003456	012767	004440	175364	MOV	#004440.	\$FPS	:SET IE BITS IN FORTRAN ANSWER
003464	005067	175362		CLR	\$FEC		:CLR FORTRAN FEC
003470	005067	175360		CLR	\$FPC		:CLR FORTRAN FPC
003474	005067	175342		CLR	FPU FPS		:CLR FPU FPS BUFFER
003500	005067	175340		CLR	FPU FEC		:CLR FPU FEC BUFFER
003504	005067	175336		CLR	FPU FPC		:CLR FPU FPC BUFFER
003510	004767	003532		GET	RANDM2		:GET RANDOM INPUT DATA
003514	004467	002560		ENTER	\$POLSH		:ENTER POLISH MODE
003520	006312			PUSH	2 WORDS	ON STACK (LONUM)	
003522	006334			PUSH	2 WORDS	ON STACK (HINUM)	
003524	000000			ADDRESS	OF FORTRAN	DIVIDE	
003526	006402			DETERMINE	THE CONDITION	CODES	
003530	006346			POP	2 WORDS	AND EXIT POLISH MODE	
003532	016700	175312		MOV	\$FPS	RC	:DISPLAY FLOATING POINT STATUS
003536	170127	040000		LDFPS	#040000		:SET FID
003542	172467	175234		LCF	LONUM.	AC0	:LOAD AC0 WITH A RANDOM NUMBER
003546	172567	175240		LCF	HINUM.	AC1	:LOAD AC1 WITH A RANDOM NUMBER
003552	172767	175254		LCF	ANS2.	AC3	:LOAD AC3 WITH THE SUM
003556	170127	004440		LDFPS	#004440		:TURN INTERRUPTS ON, EXCEPT OVERFLOW AND UNDERFLOW
003562	012767	003570	005412	MOV	#.+6.	LAD	:RESET LOOP ADDRESS

003570	172600			LCF	AC0.	AC2	:LOAD AC0 INTO AC2
003572	174601			DIVF	AC1.	AC2	:DIVIDE AC1 INTO AC2
003574	005767	175250		TEST	\$FPS		:CHECK FOR FORTRAN FPS ERROR FLAG
003600	100412			BMI	ERR7		:BRANCH IF ERROR FLAG SET
003602	170267	175234		STFPS	FPS		:STORE FLOATING POINT STATUS
003606	026767	175230	175234	CMF	FPS	\$FPS	:CHECK FPS
003614	001427			BEG	+5+		:BRANCH IF OK
003616	174267	175200		STD	AC2.	ANS1	:SAVE FPU ANSWER
003622	104001			HLT+1			:FPS ERROR
003624	000461			BR	END7		:SKIP COMPARE
003626	170000			ERR7:	CFCC		:WAIT FOR FPU TO FINISH

MACY11 27(732) 16-SEP-76
MACY11 27(732) 16-SEP-76

FLOATING POINT DIVIDE EXERCISER TEST SECTION

```

003630 026767 175206 175212      CMP      FPS,$FPS      :CHECK THE FLOATING POINT STATUS
003636 001402      BEQ      .+6      :BRANCH IF OK
003640 104377      HLT+377      :FPS ERROR
003642 000452      BR      END7      :SKIP TO END

003644 026767 175174 175200      CMP      FEC,$FEC      :CHECK THE FLOATING EXCEPTION CODES
003652 001402      BEQ      .+6      :BRANCH IF OK
003654 104377      HLT+377      :FEC IS WRONG
003656 000444      BR      END7      :SKIP TO END

003660 026767 175162 175166      CMP      FPC,$FPC      :CHECK FLOATING PC
003666 001402      BEQ      .+6      :BRANCH IF OK
003670 104377      HLT+377      :WRONG ADDRESS IN FPC
003672 000436      BR      END7      :SKIP TO END

003674 032767 000002 175146 TST7: BIT      #2,$FPS      :CHECK FOR OVERFLOW
003702 001032      BNE      END7      :BRANCH IF OVERFLOW
003704 173702      CMPE      AC2, AC3      :COMPARE FPU ANSWER TO FORTRAN ANSWER
003706 170000      CFCC      :COPY FLOATING CONDITION CODES
003710 001427      BEQ      END7      :ANSWERS CHECK
:COMPENSATE FOR FORTRAN INACCURACIES.
003712 174267 175104      STF      AC2, ANS1      :SAVE FPU ANSWER
003716 062767 000001 175100      ADD      #1, ANS1+2      :INCREMENT FPU ANSWER
003724 005567 175072      ADC      ANS1, AC3      :CHECK ANSWERS AGAIN
003730 173767 175066      CMPE      ANS1, AC3      :COPY FLOATING CONDITION CODES
003734 170000      CFCC      :BRANCH IF OK
003736 001414      BEQ      END7      :DECREMENT FPU ANSWER
003740 162767 000002 175056      SUB      #2, ANS1+2
003746 005667 175050      SBC      ANS1, AC3      :CHECK ANSWERS AGAIN
003752 173767 175044      CMPE      ANS1, AC3      :COPY FLOATING CONDITION CODES
003756 170000      CFCC      :BRANCH IF OK
003760 001403      BEQ      END7      :SAVE FPU ANSWER
003762 174267 175034      STF      AC2, ANS1      :FPU AND FORTRAN DISAGREE
003766 104000      HLT

003770 005067 175046      ENO7: CLR      FPS      :CLR FPU FPS BUFFER
003774 104400      SCOPE

```

```

*****
:TEST 10: EXERCISE DIVD (DIVIDE DOUBLE PRECISION)
: OVERFLOW AND UNDERFLOW INTERRUPTS OFF
: TRUNCATE MODE
*****

```

```

003776 012767 004640 175044      MOV      #004640,$FPS      :SET IE BITS IN FORTRAN ANSWER
004004 005067 175042      CLR      $FEC      :CLR FORTRAN FEC
004010 005067 175040      CLR      $FPC      :CLR FORTRAN FPC
004014 005067 175022      CLR      FPS      :CLR FPU FPS BUFFER
004020 005067 175020      CLR      FEC      :CLR FPU FEC BUFFER
004024 005067 175016      CLR      FPC      :CLR FPU FPC BUFFER
004030 004767 003350      JSR      PC, RANDM4      :GET RANDOM INPUT DATA
004034 004467 002240      JSR      R4, $POLSH      :ENTER POLISH MODE
004040 006302      SP,4A      :PUSH 4 WORDS ON STACK (LONUM)
004042 006324      SP,4B      :PUSH 4 WORDS ON STACK (HNUM)
004044 000000      MOVD      :ADDRESS OF FORTRAN DIVIDE
004046 006402      B7S      :DETERMINE THE CONDITION CODES

```

```

004050 006360          SPOP4X          ;POP 4 WORDS AND EXIT POLISH MODE
004052 016700 174772      MOV      $FPS,      PC      ;DISPLAY FLOATING POINT STATUS
004056 170127 040200      LDFPS     #040200          ;SET FID AND FD
004062 172467 174714      LDD       LONUM,      ACC     ;LOAD ACC WITH A RANDOM NUMBER
004066 172567 174720      LDD       HINUM,      AC1     ;LOAD AC1 WITH A RANDOM NUMBER
004072 172767 174734      LDD       ANS2,       AC3     ;LOAD AC3 WITH THE SUM
004076 170127 004640      LDFPS     #004640          ;TURN INTERRUPTS ON, EXCEPT OVER AND UNDERFLOW
004102 012767 004110 005072      MOV      #.+6,      LAD     ;RESET LOOP ADDRESS

```

```

004110 172600          LDD       ACC,      AC2      ;LOAD ACC INTO AC2
004112 174601          DIVD      AC1,      AC2      ;DIVIDE AC1 INTO AC2
004114 005767 174730      TST      $FPS          ;CHECK FOR FORTRAN FPS ERROR FLAG
004120 100412          BMI      ERR10          ;BRANCH IF ERROR FLAG SET
004122 170267 174714      STFPS     FPS          ;STORE FLOATING POINT STATUS
004126 026767 174710 174714      CMP      FPS,      $FPS     ;CHECK FPS
004134 001427          BEQ      TST10          ;BRANCH IF OK
004136 174267 174660      STD       AC2,      ANS1     ;SAVE FPU ANSWER
004142 104001          HLT+1          ;FPS ERROR
004144 000471          BR         END10          ;SKIP COMPARE

004146 170000          CFCC          ;WAIT FOR FPU TO FINISH
004150 026767 174666 174672      CMP      FPS,$FPS     ;CHECK THE FLOATING POINT STATUS
004156 001402          BEQ      .+6          ;BRANCH IF OK
004160 104377          HLT+377        ;FPS ERROR
004162 000462          BR         END10          ;SKIP TO END

004164 026767 174654 174660      CMP      FEC,$FEC     ;CHECK THE FLOATING EXCEPTION CODES
004172 001402          BEQ      .+6          ;BRANCH IF OK
004174 104377          HLT+377        ;FEC IS WRONG
004176 000454          BR         END10          ;SKIP TO END

004200 026767 174642 174646      CMP      FPC,$FPC     ;CHECK FLOATING PC
004206 001450          BEQ      END10          ;BRANCH IF OK
004210 104377          HLT+377        ;WRONG ADDRESS IN FPC
004212 000446          BR         END10          ;SKIP TO END

004214 032767 000002 174626      TST10:  BIT      #2,      $FPS     ;CHECK FOR OVERFLOW
004222 001042          BNE      END10          ;BRANCH IF OVERFLOW
004224 173702          CMPD      AC2,      AC3     ;COMPARE FPU ANSWER TO FORTRAN ANSWER
004226 170000          CFCC          ;COPY FLOATING CONDITION CODES
004230 001437          BEQ      END10          ;ANSWERS CHECK
                                ;COMPENSATE FOR FORTRAN INACCURACIES.
004232 174267 174564      STD       AC2,      ANS1     ;SAVE FPU ANSWER
004236 062767 000001 174564      ADD      #1,      ANS1+6 ;INCREMENT FPU ANSWER
004244 005567 174556      ADC      ANS1+4
004250 005567 174550      ADC      ANS1+2
004254 005567 174542      ADC      ANS1
004260 173767 174536      CMPD      ANS1,      AC3     ;CHECK ANSWERS AGAIN
004264 170000          CFCC          ;COPY FLOATING CONDITION CODES
004266 001420          BEQ      END10          ;BRANCH IF OK
004270 162767 000002 174532      SUB      #2,      ANS1+6 ;DECREMENT FPU ANSWER
004276 005667 174524      SBC      ANS1+4
004302 005667 174516      SBC      ANS1+2

```

K02

MAINDEC-11-CCFPU-D
CCFPU.D.P11FLOATING POINT DIVIDE EXERCISER
TEST SECTION

MACY11 27(732) 16-SEP-76 16:30 PAGE 23

```

004306 005667 174510 SBC ANS1
004312 173767 174504 CMPD ANS1, AC3 ;CHECK ANSWERS AGAIN
004316 170000 CFCC ;COPY FLOATING CONDITION CODES
004320 001403 BEQ END10 ;BRANCH IF OK
004322 174267 174474 STD AC2, ANS1 ;SAVE FPU ANSWER
004326 104000 HLT ;FPU AND FORTRAN DISAGREE

004330 005067 174506 END10: CLR FPS ;CLR FPU FPS BUFFER
004334 104400 SCOPE

```

```

:*****
:TEST 11: EXERCISE DIVF (DIVIDE FLOATING,
:          INTERRUPT DISABLE SET
:          ROUNDING MODE
:*****

```

```

004336 012767 047400 174504 MOV #047400, $FPS ;SET IE BITS IN FORTRAN ANSWER
004344 005067 174532 CLR $FEC ;CLR FORTRAN FEC
004350 005067 174530 CLR $FPC ;CLR FORTRAN FPC
004354 005067 174462 CLR FPS ;CLR FPU FPS BUFFER
004360 005067 174460 CLR FEC ;CLR FPU FEC BUFFER
004364 005067 174456 CLR FPC ;CLR FPU FPC BUFFER
004370 004767 002652 JSR PC, RANDM2 ;GET RANDOM INPUT DATA
004374 004467 001700 JSR R4, $POLSH ;ENTER POLISH MODE
004400 006312 $P.2A ;PUSH 2 WORDS ON STACK (LONUM)
004402 006334 $P.2B ;PUSH 2 WORDS ON STACK (MINUM)
004404 000000G $DVR ;ADDRESS OF FORTRAN DIVIDE
004406 006402 $TST ;DETERMINE THE CONDITION CODES
004410 006346 $POP2X ;PCP 2 WORDS AND EXIT POLISH MODE

004412 016700 174432 MOV $FPS, R0 ;DISPLAY FLOATING POINT STATUS
004416 170127 040000 LD FPS, #040000 ;SET FID
004422 172467 174354 LDF LONUM, AC0 ;LOAD AC0 WITH A RANDOM NUMBER
004426 172567 174360 LDF MINUM, AC1 ;LOAD AC1 WITH A RANDOM NUMBER
004432 172767 174374 LDF ANS2, AC3 ;LOAD AC3 WITH THE SUM
004436 170127 047400 LD FPS, #047400 ;SET INTERRUPT DISABLE AND INTERRUPT BITS
004442 012767 004450 004532 MOV #.+6, LAD ;RESET LOOP ADDRESS

```

:*****

```

004450 172600 RET11: LDF AC0, AC2 ;LOAD AC0 INTO AC2
004452 174601 DIVF AC1, AC2 ;DIVIDE AC1 INTO AC2
004454 170267 174362 STFPS FPS ;STORE FLOATING POINT STATUS
004460 005767 174364 TST $FPS ;CHECK FOR ERROR FLAG
004464 100410 BMI ERR11 ;BRANCH IF ERROR FLAG SET
004466 026767 174350 174354 CMP FPS, $FPS ;CHECK FPS
004474 001430 BEQ TST11 ;BRANCH IF OK
004476 174267 174320 STF AC2, ANS1 ;SAVE FPU ANSWER
004502 104001 HLT+1 ;FPS ERROR
004504 000451 BR END11 ;SKIP COMPARE

004506 170367 174332 ERR11: STST FEC AND FPC ;STORE FEC AND FPC
004512 026767 174324 174330 CMP FPS, $FPS ;CHECK THE FLOATING POINT STATUS
004520 001402 BEQ .+6 ;BRANCH IF OK
004522 104377 HLT+377 ;FPS ERROR
004524 000441 BR END11 ;SKIP TO END

```

004526	026767	174312	174316	CMP	FEC, \$FEC		:CHECK THE FLOATING EXCEPTION CODES
004534	001402			BEQ	+.6		:BRANCH IF OK
004536	104377			HLT+377			:FEC IS WRNG
004540	000433			BR	END11		:SKIP TO END
004542	026767	174300	174304	CMP	FPC, \$FPC		:CHECK FLOATING PC
004550	001402			BEQ	TST11		:BRANCH IF OK
004552	104377			HLT+377			:WRONG ADDRESS IN FPC
004554	000425			BR	END11		:SKIP TO END
004556	032767	000002	174264	TST11:	BIT	#2, \$FPS	:CHECK FOR OVERFLOW
004564	001021			BNE	END11		:BRANCH IF OVERFLOW
004566	173702			CMPF	AC2, AC3		:COMPARE FPU ANSWER TO FORTRAN ANSWER
004570	170000			CFCC			:COPY FLOATING CONDITION CODES
004572	001416			BEQ	END11		:ANSWERS CHECK
					:COMPENSATE FOR FORTRAN		:INACCURACIES.
004574	174267	174222		STF	AC2, ANS1		:SAVE FPU ANSWER
004600	162767	000001	174216	SUB	#1, ANS1+2		:DECREMENT FPU ANSWER
004606	005667	174210		SBC	ANS1		
004612	173767	174204		CMPF	ANS1, AC3		:CHECK ANSWERS AGAIN
004616	170000			CFCC			:COPY FLOATING CONDITION CODES
004620	001403			BEQ	END11		:BRANCH IF OK
004622	174267	174174		STF	AC2, ANS1		:SAVE FPU ANSWER
004626	104000			HLT			:FPU AND FORTRAN DISAGREE
004630	005067	174236		END11:	CLR	FPS	:CLR FPU FPS BUFFER
004634	104400			SCOPE			

```

*****
:TEST 12:      EXERCISE DIVD (DIVIDE DOUBLE PRECISION)
:              INTERUPT DISABLE SET
:              ROUNDING MODE
*****

```

004636	012767	047600	174204	MOV	#047600,\$FPS	:SET FID AND IE BITS IN FORTRAN ANSWER
004644	005067	174202		CLR	\$FEC	:CLR FORTRAN FEC
004650	005067	174200		CLR	\$FPC	:CLR FORTRAN FPC
004654	005067	174162		CLR	FPS	:CLR FPU FPS BUFFER
004660	005067	174160		CLR	FEC	:CLR FPU FEC BUFFER
004664	005067	174156		CLR	FPC	:CLR FPU FPC BUFFER
004670	004767	002510		JSR	PC, RANDM4	:GET RANDOM INPUT DATA
004674	004467	001400		JSR	R4, \$POLSH	:ENTER POLISH MODE
004700	006302			\$P.4A		:PUSH 4 WORDS ON STACK (LONUM)
004702	006324			\$P.4B		:PUSH 4 WORDS ON STACK (MINUM)
004704	000000G			\$DIVD		:ADDRESS OF FORTRAN DIVIDE
004706	006402			\$TST		:DETERMINE THE CONDITION CODES
004710	006360			\$POP4X		:PCP 4 WORDS AND EXIT POLISH MODE
004712	016700	174132		MOV	\$FPS, R0	:DISPLAY FLOATING POINT STATUS
004716	170127	040200		LDFPS	#040200	:SET FID AND FD
004722	172467	174054		LDD	LONUM, ACO	:LOAD ACO WITH A RANDOM NUMBER
004726	172567	174060		LDD	MINUM, AC1	:LOAD AC1 WITH A RANDOM NUMBER
004732	172767	174074		LDD	ANS2, AC3	:LOAD AC3 WITH THE SUM
004736	170127	047600		LDFPS	#047600	:SET INTERUPT DISABLE AND INTERUPT BITS
004742	012767	004750	004232	MOV	#.+6, LAC	:RESET LOOP ADDRESS

004750	172600			LDD	ACO, AC2	:LOAD ACO INTO AC2
004752	174601			DIVD	AC1, AC2	:DIVIDE AC1 INTO AC2
004754	170267	174062		STFPS	FPS	:STORE FLOATING POINT STATUS
004760	005767	174064		TST	\$FPS	:CHECK FOR ERROR FLAG
004764	100410			BMI	EPR12	:BRANCH IF ERROR FLAG SET
004766	026767	174050	174054	CMP	FPS, \$FPS	:CHECK FPS
004774	001430			BEQ	TST12	:BRANCH IF OK
004776	174267	174020		STD	AC2, ANS1	:SAVE FPU ANSWER
005002	104001			HLT+1		:FPS ERROR
005004	000455			BR	END12	:SKIP COMPARE
005006	170367	174032		STST	FEC	:STORE FEC AND FPC
005012	026767	174024	174030	CMP	FPS, \$FPS	:CHECK THE FLOATING POINT STATUS
005020	001402			BEQ	+.6	:BRANCH IF OK
005022	104377			HLT+377		:FPS ERROR
005024	000445			BR	END12	:SKIP TO END
005026	026767	174012	174016	CMP	FEC,\$FEC	:CHECK THE FLOATING EXCEPTION CODES
005034	001402			BEQ	+.6	:BRANCH IF OK
005036	104377			HLT+377		:FEC IS WRONG
005040	000437			BR	END12	:SKIP TO END
005042	026767	174000	174004	CMP	FPC,\$FPC	:CHECK FLOATING PC
005050	001402			BEQ	TST12	:BRANCH IF OK
005052	104377			HLT+377		:WRONG ADDRESS IN FPC

```

005054 000431          BR      END12          ;SKIP TO END
005056 032767 000002 173764 TST12: BIT      #2      $FPS      ;CHECK FOR OVERFLOW
005064 001025          BNE      END12          ;BRANCH IF OVERFLOW
005066 173702          CMPD     AC2,      AC3      ;COMPARE FPU ANSWER TO FORTRAN ANSWER
005070 170000          CFCC          ;COPY FLOATING CONDITION CODES
005072 001422          BEQ      END12          ;ANSWERS CHECK
          ;COMPENSATE FOR FORTRAN INACCURACIES.
005074 174267 173722          STD      AC2,      ANS1      ;SAVE FPU ANSWER
005100 162767 000001 173722          SUB      #1,      ANS1+6 ;DECREMENT FPU ANSWER
005106 005667 173714          SBC      ANS1+4
005112 005667 173706          SBC      ANS1+2
005116 005667 173700          SBC      ANS1
005122 173767 173674          CMPD     ANS1,      AC3      ;CHECK ANSWERS AGAIN
005125 170000          CFCC          ;COPY FLOATING CONDITION CODES
005130 001403          BEQ      END12          ;BRANCH IF OK
005132 174267 173664          STC      AC2,      ANS1      ;SAVE FPU ANSWER
005136 104000          HLT          ;FPU AND FORTRAN DISAGREE

005140 005067 173676          END12: CLR      FPS          ;CLR FPU FPS BUFFER
005144 104400          SCOPE

```

```

:*****
:TEST 13:      EXERCISE DIVF (DIVIDE FLOATING)
:              INTERRUPT DISABLE SET
:              TRUNCATE MODE
:*****

```

```

005146 012767 047440 173674          MOV      #047440,$FPS ;SET FID AND IE BITS IN FORTRAN ANSWER
005154 005067 173672          CLR      $FEC      ;CLR FORTRAN FEC
005160 005067 173670          CLR      $FPC      ;CLR FORTRAN FPC
005164 005067 173652          CLR      FPS      ;CLR FPU FPS BUFFER
005170 005067 173650          CLR      FEC      ;CLR FPU FEC BUFFER
005174 005067 173646          CLR      FPC      ;CLR FPU FPC BUFFER
005200 004767 002042          JSR      PC,      RANDM2 ;GET RANDOM INPUT DATA
005204 004467 001070          JSR      R4,      $POLSH ;ENTER POLISH MODE
005210 006312          $P.2A          ;PUSH 2 WORDS ON STACK (LONUM)
005212 006334          $P.2B          ;PUSH 2 WORDS ON STACK (HINUM)
005214 000000G          $DVR          ;ADDRESS OF FORTRAN DIVIDE
005216 006402          $TST          ;DETERMINE THE CONDITION CODES
005220 006346          $POP2X         ;POP 2 WORDS AND EXIT POLISH MODE

005222 016700 173622          MOV      $FPS,      R0      ;DISPLAY FLOATING POINT STATUS
005226 170127 040000          LDFPS     #040000          ;SET FID
005232 172467 173544          LDF      LONUM,      ACO      ;LOAD ACO WITH A RANDOM NUMBER
005236 172567 173550          LDF      HINUM,      AC1      ;LOAD AC1 WITH A RANDOM NUMBER
005242 172767 173564          LDF      ANS2,      AC3      ;LOAD AC3 WITH THE SUM
005246 170127 047440          LDFPS     #047440          ;SET INTERRUPT DISABLE AND INTERRUPT BITS
005252 012767 005260 003722          MOV      #.+6,      LAD      ;RESET LOOP ADDRESS

```

[illegible]

FLOATING POINT DIVIDE EXERCISER
TEST SECTION

TEST 14: EXERCISE DIVD (DIVIDE DOUBLE PRECISION)
INTERUPT DISABLE SET
TRUNCATE MODE

005540	004767	047640	173352	MOV	8047640, SFPS	:SET FID AND IE BITS IN FORTRAN ANSWER
005541	005367	173350		CLR	SFEC	:CLR FORTRAN FEC
005542	005367	173346		CLR	SFPC	:CLR FORTRAN FPC
005543	005367	173330		FPS		:CLR FPU FPS BUFFER
005544	005367	173326		FEC		:CLR FPU FEC BUFFER
005545	005367	173324		FPC		:CLR FPU FPC BUFFER
005546	004767	001656		GET	RANDM4	:GET RANDOM INPUT DATA
005547	004767	000546		ENTER	POLISH	:ENTER POLISH MODE
005548	006302			PUSH	4 WORDS ON STACK (LONUM	:PUSH 4 WORDS ON STACK (LONUM
005549	006324			PUSH	4 WORDS ON STACK (FINUM	:PUSH 4 WORDS ON STACK (FINUM
005550	000000			ADDRESS	OF FORTRAN DIVIDE	:ADDRESS OF FORTRAN DIVIDE
005551	006402			DETERMINE	THE CONDITION CODES	:DETERMINE THE CONDITION CODES
005552	006350			POP	4 WORDS AND EXIT POLISH MODE	:POP 4 WORDS AND EXIT POLISH MODE
005553	016700	173300		MOV	SFPS, PC	:DISPLAY FLOATING POINT STATUS
005554	170127	040200		SET	FID AND FD	:SET FID AND FD
005555	172467	173300		LOAD	AC0 WITH A RANDOM NUMBER	:LOAD AC0 WITH A RANDOM NUMBER
005556	172557	173300		LOAD	AC1 WITH A RANDOM NUMBER	:LOAD AC1 WITH A RANDOM NUMBER
005557	172767	173300		LOAD	AC3 WITH THE SUM	:LOAD AC3 WITH THE SUM
005558	173300	173300		SET	INTERUPT DISABLE AND INTERUPT BITS	:SET INTERUPT DISABLE AND INTERUPT BITS
005559	173300	003400		RESET	LOOP ADDRESS	:RESET LOOP ADDRESS

005602	172600			LOAD	AC0 INTO AC2	:LOAD AC0 INTO AC2
005603	174507			DIVIDE	AC1 INTO AC2	:DIVIDE AC1 INTO AC2
005604	170267	173230		STORE	FLOATING POINT STATUS	:STORE FLOATING POINT STATUS
005605	005767	173232		CHECK	FOR ERROR FLAG	:CHECK FOR ERROR FLAG
005606	100410			BRANCH	IF ERROR FLAG SET	:BRANCH IF ERROR FLAG SET
005607	026767	173216	173222	CHECK	FPS	:CHECK FPS
005608	001430			BRANCH	IF OK	:BRANCH IF OK
005609	174267	173166		SAVE	FPU ANSWER	:SAVE FPU ANSWER
005610	104001			FPS	ERROR	:FPS ERROR
005611	000472			SKIP	COMPARE	:SKIP COMPARE
005612	170367	173200		STORE	FEC AND FPC	:STORE FEC AND FPC
005613	026767	173172	173176	CHECK	THE FLOATING POINT STATUS	:CHECK THE FLOATING POINT STATUS
005614	001402			BRANCH	IF OK	:BRANCH IF OK
005615	104377			FPS	ERROR	:FPS ERROR
005616	000462			SKIP	TO END	:SKIP TO END
005617	026767	173160	173164	CHECK	THE FLOATING EXCEPTION CODES	:CHECK THE FLOATING EXCEPTION CODES
005618	001402			BRANCH	IF OK	:BRANCH IF OK
005619	104377			FEC	IS WRONG	:FEC IS WRONG
005620	000454			SKIP	TO END	:SKIP TO END
005621	026767	173146	173152	CHECK	FLOATING PC	:CHECK FLOATING PC
005622	001402			BRANCH	IF OK	:BRANCH IF OK
005623	104377			WRONG	ADDRESS IN FPC	:WRONG ADDRESS IN FPC

003

MACY11-20FPL-2
MACY11.F11

FLOATING POINT DIVIDE EXERCISE
TEST SECTION

MACY11 27(732) 16-SEP-76 16:30 PAGE 29

005706	000446		BR	END14		:SKIP TO END		
005710	032767	000002	173132	TST14:	BIT	B2	SFPS	:CHECK FOR OVERFLOW
005716	001243				BNE	END14		:BRANCH IF OVERFLOW
005720	173703				CMPO	AC2.	AC3	:COMPARE FPU ANSWER TO FORTRAN ANSWER
005724	170000				CFCC			:COPY FLOATING CONDITION CODES
005728	001437				BEG	END14		:ANSWERS CHECK
005732	174267	173070			STO	AC2.	ANS1	:SAVE FPU ANSWER
005736	062767	000001	173070		ADD	B1	ANS1+6	:INCREMENT FPU ANSWER
005740	005567	173062			ADD	ANS1+4		
005744	005567	173054			ADD	ANS1+2		
005750	005567	173046			ADD	ANS1		
005754	173767	173042			CMPO	ANS1.	AC3	:CHECK ANSWERS AGAIN
005760	170000				CFCC			:COPY FLOATING CONDITION CODES
005764	001420				BEG	END14		:BRANCH IF OK
005768	162767	000002	173036		B2	ANS1+6		:DECREMENT FPU ANSWER
005772	005667	173030			ANS1+4			
005776	005667	173022			ANS1+2			
006002	005667	173014			ANS1			
006006	173767	173010			ANS1.	AC3		:CHECK ANSWERS AGAIN
006012	170000				CFCC			:COPY FLOATING CONDITION CODES
006016	001420				BEG	END14		:BRANCH IF OK
006020	174267	173000			AC2.	ANS1		:SAVE FPU ANSWER
006024	104000				END14			:FPU AND FORTRAN DISAGREE
006028	005067	173012		END14:	CLR	FPS		:CLR FPU FPS BUFFER
006032	004400				SCOPE			

E03

MACY11 27(732) 16-SEP-76 16:30 PAGE 30

MAYNCE-11-00FL-0
00000000FLOATING POINT DIVIDE EXERCISER
SELL AND SCOPE ROUTINE

002733	032737	002000	177570	DONE:	BIT	8SW10,28SW	:RING THE BELL?
002734	001005				BNE	18	:NO!
002735	012767	000207	003126		MOV	8BELL...TYPE	:TYPE A BELL
002736	000004	011176			TYPE	-TYPE	
002737	005046			18:	CLR	-(6)	:CLEAR TRACE TRAP
002738	032737	010000	177570		BIT	8SW12,28SW	:RUN WITH TR?
002739	001010				BNE	28	
002740	005167	003106			COM	TRPB	
002741	100005				BPL	28	
002742	052716	000020			BIS	820,6)	:SET TRACE TRAP
002743	012746	006132			MOV	835, - 6	:JUMP TO START OF TEST
002744	000002				RTI		
002745	012746	006114		28:	MOV	845, - 6	:JUMP TO START OF TEST
002746	000002				RTI		
002747	013700	000042		48:	MOV	8842,PC	:GET MONITOR ADDRESS
002748	001404				BEC	38	:IF NONE
002749	004710				JSR	7,0)	:GO TO MONITOR
002750	000240				NOP		
002751	000240				NOP		
002752	000240				NOP		
002753	000137	000200		38:	JMP	28200	:JUMP TO START OF TEST
002754	000002			YESRT:	RTI		:RETURN TO PROGRAM FROM TRAP
002755	032737	000400	177570	.ENT:	BIT	8SW08,28SW	:KILL LOUB OR LOOP ON SPEC. TEST
002756	001404				BEO	18	
002757	123767	177570	172622		CMPB	28SW,ICNT	:ON RIGHT TEST *SW7-0*
002758	001437				BEO	OVER	
002759	113703	177570		18:	MOVB	28SW,R3	:GET US BITS
002760	170003				LOUB		
002761	032737	040000	177570		BIT	8SW14,28SW	:LOOP ON TEST
002762	001026				BNE	KIT	
002763	032737	004000	177570		BIT	8SW11,28SW	:KILL ITERATIONS
002764	001012				BNE	SAVLAD	
002765	105767	172567			TSIB	ICNT+1	
002766	001404				BEO	28	:BRANCH IF FIRST
002767	126767	002770	172557		CMPB	TIMES,ICNT+1	:DONE?
002768	001013				BNE	KIT	:BRANCH IF NOT
002769	112767	000001	172547	28:	MOVB	81,ICNT+1	:FIRST ITERATION
002770	105267	172542		SAVLAD:	INCB	ICNT	:COUNT TEST NUMBERS
002771	011667	002740			MOV	(6),LAD	:SAVE LOOP ADDRESS
002772	016737	172532	177570		MOV	ICNT,28DISPLAY	:DISPLAY TEST NO. AND ITERATION COUNT
002773	000002				RTI		:RETURN
002774	105267	172523		KIT:	INCB	ICNT+1	
002775	016737	172516	177570	0.EP:	MOV	ICNT,28DISPLAY	:SET UP DISPLAY
002776	005767	002712			TSIB	LAD	:FIRST ONE
002777	000002				BEO	SAVLAD	
002778	000002	002704			LAD	LAD, 6	:FUDGE RETURN ADDRESS
002779	000002				RTI		:FUDGE RETURN ADDRESS

MACY11 27 732. 16-SEP-76
16:30 PAGE 31

FLOATING POINT DIVIDE EXERCISER
FORTRAN ROUTINES (POLISH MODE)

006322	000134			\$FOLSH: JMP	2(R4)+	
006322	016746	172502		\$P.4A: MOV	LONUM+6, - SP	
006326	016746	172474		MOV	LONUM+4, - SP	
006326	016746	172466		\$F.2A: MOV	LONUM+2, - SP	
006326	016746	172460		MOV	LONUM, - SP	
006326	000134			JMP	2(R4)+	
006324	016746	172470		\$P.4B: MOV	HINUM+6, - SP	
006330	016746	172462		MOV	HINUM+4, - SP	
006334	016746	172454		\$P.2B: MOV	HINUM+2, - SP	
006340	016746	172446		MOV	HINUM, - SP	
006344	000134			JMP	2(R4)+	
006346	012667	172460		\$PCP2Y: MOV	SP, +, ANS2	
006352	012667	172456		MOV	SP, +, ANS2+2	
006356	000204			RTS	R4	:EXIT POLISH MODE
006360	012667	172446		\$PCP4Y: MOV	SP, +, ANS2	
006364	012667	172444		MOV	SP, +, ANS2+2	
006370	012667	172442		MOV	SP, +, ANS2+4	
006374	012667	172440		MOV	SP, +, ANS2+6	
006400	000204			RTS	R4	:EXIT POLISH MODE
006402	032767	100002	172440	\$TER: BTT	#100002, SFPS	:CHECK FOR ERROR FLAG AND OVERFLOW
006410	100453			\$TER		:BRANCH IF FLAG SET
006412	001061			\$TCV		:BRANCH IF OVERFLOW
006414	032716	077600		BTT	#077600, (6)	:TEST THE EXPONENT
006420	001041			BNE	STST1	:BRANCH IF NOT ZERO
006422	052767	000004	172420	BTT	#04, SFPS	:SET Z BIT
006430	032767	077600	172354	BTT	#077600, HINUM	:CHECK DIVISOR FOR ZERO
006436	001032			BNE	STST1	:BRANCH IF NON-ZERO
006440	052767	100000	172402	BIS	#100000, SFPS	:SET THE ERROR FLAG
006446	012767	000004	172376	MOV	#04, SFEC	:DIVIDE BY ZERO EXCEPTION CODE
006454	116701	172320		MOVB	ICNT, R1	:GET THE TEST NUMBER
006460	006301			ASL	R1	:#2
006462	016167	006572	172364	MOV	RETAD-2(R1), SFPC	:GET EXPECTED PC
006470	016716	172306		MOV	LONUM, (SP)	:RESTORE DIVIDEND
006474	016766	172304	000002	MOV	LONUM+2, 2(SP)	
006502	105767	172342		TSTB	SFPS	:CHECK FOR DOUBLE PRECISION
006506	100006			BPL	STST1	:BRANCH IF NO
006510	016766	172272	000004	MOV	LONUM+4, 4(SP)	
006516	016766	172266	000006	MOV	LONUM+6, 6(SP)	
006524	005716			STST1: TST	(SP)	:FIND THE SIGN
006526	100003			BPL	STST2	:BRANCH IF PLUS
006530	052767	000010	172312	BIS	#10, SFPS	:SET N BIT
006536	000134			STST2: JMP	2(R4)+	
006540	116701	172234		\$TER: MOVB	ICNT, R1	:GET TEST NUMBER
006544	005301			DEC	R1	:SET POINTER
006546	006301			ASL	R1	:TEST # * 2
006550	016167	006574	172276	MOV	RETAD(1), SFPC	:STORE FORTRAN FPC
006556	022626			STCV: CMP	SP, +, (SP)+	:POP 2 WORDS
006560	105767	172264		TSTB	SFPS	:CHECK FC BIT
006564	100001			BPL	STCV1	:BRANCH IF FLOATING MODE

CONFIDENTIAL

TRAINING POINT DIVIDE EXERCISER

[illegible]

STOV1: CMP (30)♦
IST (4)♦
RTS 04

```
SP, + : "POP" 2 MORE WORDS
      : SKIP "POPX" ROUTINE
      : EXIT POLISH MODE
```

0006574	0001342
0006576	0001630
0006600	0002126
0006622	0002426
0006624	0002732
0006636	0003244
0006610	0003572
0006612	0004112
0006614	0004458
0006616	0004798
0006620	0005262
0006622	0005604

[illegible]

006624	020027	002405
006630	001420	
006632	020027	004005
006636	001433	
006640	020027	003500
006644	001454	
006646	020027	002053
006652	001463	
006654	020027	004003
006660	001531	
006662	020027	001403
006666	001534	
006670	000000	

```

SERPP:  CMP      RC,      #2405      :CHECK FOR UNDERFLOW.  FD=0
        BEQ     $UNDR0    :BRANCH IF UNDERFLOW
        CMP     RC,      #4005      :CHECK FOR UNDERFLOW.  FD=1
        BEQ     $UNDR1    :BRANCH IF UNDERFLOW
        CMP     RC,      #3003      :CHECK FOR OVERFLOW.  FD=0
        BEQ     $OVER0    :BRANCH IF OVERFLOW
        CMP     RC,      #2003      :CHECK FOR OVERFLOW.  FD=1
        BEQ     $OVER1    :BRANCH IF OVERFLOW
        CMP     RC,      #4003      :CHECK FOR DIVIDE BY ZERO.  FD=0
        BEQ     $DVBZ0    :BRANCH IF DIVIDE BY ZERO
        CMP     RC,      #1403      :CHECK FOR DIVIDE BY ZERO.  FD=1
        BEQ     $DVBZ1    :BRANCH IF DIVIDE BY ZERO
SUNKER:  HALT     :UNKNOWN ERROR!

```

006672	032767	003000
0066700	001476	
006702	004457	177372
006706	006312	
006710	006334	
006712	007100	
006714	0000006	
006716	007122	
006720	006402	
006722	006346	
006724	000415	

```

$UNDR0: BIT      #003000,$FPS      :CHECK FOR INTERRUPTS ON OR OFF
          BEQ      $FRTS           :BRANCH IF OFF
          JSR      R4,$POLSH       :ENTER POLISH MODE
          SP,2A                    :PUSH 2 WORDS ON STACK (LONUM)
          SP,2B                    :PUSH 2 WORDS ON STACK (HINUM)
          $SB200                   :SUBTRACT 200 FROM THE EXPONENT OF THE DIVERGE
          $DVR                     :ADDRESS OF FORTRAN DIVIDE
          $200SB                   :SUBTRACT 200 FROM THE EXPONENT OF $VS
          $TST                     :DETERMINE CONDITION CODES
          $POP2X                   :POP 2 WORDS AND EXIT POLISH MODE
          BR      $UNDR0

```

006726	032767	003000
006734	001460	
006736	004467	177336
006742	006302	
006744	006324	
006746	007100	
006750	000000G	
006752	007122	
006754	006402	
006756	006360	
006760	052767	100000
006762	012767	100012
006764	000000	
006766	000000	
006768	000000	
006770	000000	
006772	000000	
006774	000000	
006776	000000	
006778	000000	
006780	000000	
006782	000000	
006784	000000	
006786	000000	
006788	000000	
006790	000000	
006792	000000	
006794	000000	
006796	000000	
006798	000000	
006800	000000	

```

$UNCP1: BIT      #002000.$FPS      :CHECK FOR INTERRUPTS ON OR OFF
        BEQ      $SERIS             :BRANCH IF OFF
        JSR      R4, $POLSH         :ENTER POLISH MODE
        SP, 4A                      :PUSH 2 WORDS ON STACK (LONUM)
        SP, 4B                      :PUSH 2 WORDS ON STACK (HINUM)
        $SB200                      :SUBTRACT 200 FROM THE EXPONENT OF THE DIVISOR
        $DIVD                       :ADDRESS OF FORTRAN DIVIDE
        $2005B                      :SUBTRACT 200 FROM THE EXPONENT OF ANS
        $T5                          :DETERMINE CONDITION CODES
        $POPF4                       :POP 4 WORDS AND EXIT POLISH MODE
$UNCP2: BIT      #100000.$FPS      :SET FPS ERROR FLAG
        $NOV      #12, $FEC         :SET UNDERFLOW EXCEPTION CODE
        $NOV      #12, $FEC         :SET UNDERFLOW EXCEPTION CODE

```

MACY11 27(732) 16-SEP-76
16:30

FLOATING POINT DIVIDE EXERCISE FORTRAN ROUTINES (POLISH MODE)

007076	004467	177276	\$OVER0:	JSR	R4,	\$POLSH	:ENTER POLISH MODE
007082	006312			\$P.2A			:PUSH 2 WORDS ON STACK (LONUM)
007084	006334			\$P.2B			:PUSH 2 WORDS ON STACK (MINUM)
007086	007114			\$AD200			:ADD 200 TO THE EXPONENT OF THE DEVI
007090	0000003			\$DVR			:ADDRESS OF FORTRAN DIVIDE
007092	007135			\$200AD			:ADD 200 TO THE EXPONENT OF ANS
007094	006402			\$TST			:DETERMINE CONDITION CODES
007096	006346			\$POP2X			:POP 2 WORDS AND EXIT POLISH MODE
007098	000411			BR	\$OVERA		
007092	004467	177252	\$OVER1:	JSR	R4,	\$POLSH	:ENTER POLISH MODE
007096	006302			\$P.4A			:PUSH 2 WORDS ON STACK (LONUM)
007098	006324			\$P.4B			:PUSH 2 WORDS ON STACK (MINUM)
007100	007114			\$AD200			:ADD 200 TO THE EXPONENT OF THE DEVI
007102	0000003			\$DVR			:ADDRESS OF FORTRAN DIVIDE
007104	007106			\$200AD			:ADD 200 TO THE EXPONENT OF ANS
007106	006402			\$TST			:DETERMINE CONDITION CODES
007108	006360			\$POP4X			:POP 4 WORDS AND EXIT POLISH MODE
007110	032767	003000	\$OVERA:	BIT	#003000, \$FPS		:CHECK FOR INTERRUPTS ON OR OFF
007112	001406			BEG	\$OVR1		:BRANCH IF OFF
007114	052767	100000		BIS	#100000, \$FPS		:SET FPS ERROR FLAG
007116	012767	000010		MOV	#10, \$FEC		:SET OVERFLOW EXCEPTION CODE
007118	052767	000002	\$OVR1:	BIS	#02, \$FPS		:SET OVERFLOW BIT IN FPS
007120	000205		\$ERTS:	RTS	%5		
007100	162716	040000	\$SB200:	SUB	#040000, \$P.		
007104	000134			JMP	2(R4)+		
007106	062716	140000	\$200AD:	ADD	#140000, (\$P.)		
007110	000134			JMP	2(R4)+		
007114	062716	040000	\$AD200:	ADD	#040000, (\$P.)		
007118	000134			JMP	2(R4)+		
007122	162716	040000	\$200SB:	SUB	#040000, (\$P.)		
007126	032767	000003		BIT	#3, PS		:TEST FOR C OR V BITS
007130	001402			BEG	\$201SB		
007134	062716	100000		ADD	#100000, (\$P.)		
007138	000134		\$201SB:	JMP	2(R4)+		
007142							
007144	004467	177130	\$DVBZ0:	JSR	R4,	\$POLSH	:ENTER POLISH MODE
007146	006312			\$P.2A			:PUSH DIVIDEND
007148	006402			\$TST			:CALCULATE CONDITION CODES
007150	006346			\$POP2X			:POP DIVIDEND INTO ANSWER
007152	000405			BR	\$DVBZA		
007156							
007160	004467	177114	\$DVBZ1:	JSR	R4,	\$POLSH	:ENTER POLISH MODE
007164	006302			\$P.4A			:PUSH DIVIDEND
007168	006402			\$TST			:CALCULATE CONDITION CODES
007172	006360			\$POP4X			:POP DIVIDEND INTO ANSWER
007176	052767	100000	\$DVBZA:	BIS	#100000, \$FPS		:SET FPS ERROR FLAG
007180	012767	000004		MOV	#04, \$FEC		:DIVIDE BY ZERO EXCEPTION CODE
007184	000205			RTS	%5		:RETURN TO FORTRAN

MACY11-11-00F1-1
00F1-1-1-1

FLOATING POINT DIVIDE EXERCISER
FORTRAN ROUTINES (POLISH MODE)

MACY11 27(732) 16-SEP-76 16:30 PAGE 34

:FPP INTERRUPT SERVICE ROUTINE

:*****

007210	170267	171626	FLTERR: STFPS	FPS	:STORE FLOATING POINT STATUS
007214	170367	171624	STST	FEC	:STORE FLOATING EXCEPTION CODES
007220	032767	040000 171622	BIT	840000, SFPS	:CHECK INTERRUPT DISABLE
007226	001402		BEQ	.+6	:BRANCH IF OFF
007230	104377		HLT+377		:INTERUPT NOT SUPPOSED TO BE ENABLED
007236	000404		BR	ERR1	
007234	005767	171610	TSI	SFPS	:CHECK FORTRAN FPS FOR ERROR FLAG
007240	100404		BNI	ERR1	:BRANCH IF FLAG SET
007246	104377		HLT+377		:FLOATING POINT STATUS ERROR
007244	000002		ERR1:	ERR1	

007246	010046		RANDM2: MOV	%0.-(6)	:SAVE R0
007250	010146		MOV	%1.-(6)	:SAVE R1
007252	010246		MOV	%2.-(6)	:SAVE R2
007254	010346		MOV	%3.-(6)	:SAVE R3
007256	010446		MOV	%4.-(6)	:SAVE R4
007260	010546		MOV	%5.-(6)	:SAVE R5
007262	012704	001002	MOV	#LONUM,%4	:SET UP LONUM POINTER
007266	012705	001012	MOV	#HINUM,%5	:SET UP HINUM POINTER
007272	011400		MOV	(4),%0	:SET R0 WITH LOW
007274	011501		MOV	(5),%1	:SET R1 WITH HIGH
007276	012703	177771	REPET2: MOV	#-7,%3	:SET SHIFT COUNT
007302	005002		CLR	%2	
007304	006300		SHIFT2: ASL	%0	:SHIFT R0 LEFT AND
007306	006101		ROL	%1	:ROTATE CARRY INTO R1 AND
007310	006102		ROL	%2	:ROTATE CARRY INTO R2
007312	005203		INC	%3	:CHECK FOR DONE
007314	001373		BNE	SHIFT2	:CONTINUE SHIFT LOOP
007316	061402		ADD	(4),%2	:ADD NUMBER TO MAKE X 129
007320	005501		ADC	%1	:PROPOGATE CARRY
007322	061501		ADD	(5),%1	:ADD NUMBER TO MAKE X 129
007324	005502		ADC	%2	:PROPOGATE CARRY
007326	062700	001057	ADD	#1057,%0	:ADD LOW CONSTANT
007332	005501		ADC	%1	:PROPOGATE CARRY
007334	005502		ADC	%2	:PROPOGATE CARRY
007336	062701	047401	ADD	#47401,%1	:ADD HIGH CONSTANT
007342	005502		ADC	%2	:PROPOGATE CARRY
007344	062702	000006	ADD	#5,%2	:ADD HIGHEST CONSTANT
007350	060200		ADD	%2,%0	:REPRIME R0 WITH HIGHEST DIGIT
007352	005501		ADC	%1	:PROPOGATE CARRY
007354	010024		MOV	%0,(4)+	:SAVE R0
007356	010125		MOV	%1,(5)+	:SAVE R1
007360	020427	001006	CMP	%4,#LONUM+4	:CHECK FOR DONE ENOUGH
007364	001344		BNE	REPET2	:BRANCH IF NOT DONE
007366	012605		MOV	(6)+,%5	:RESTORE R5
007370	012604		MOV	(6)+,%4	:RESTORE R4
007372	012603		MOV	(6)+,%3	:RESTORE R3
007374	012602		MOV	(6)+,%2	:RESTORE R2
007376	012601		MOV	(6)+,%1	:RESTORE R1
007400	012600		MOV	(6)+,%0	:RESTORE R0
007402	000207		RTS		:RETURN

007404	010046		RANDM4: MOV	%0, -(6)	:SAVE R0
007406	010146		MOV	%1, -(6)	:SAVE R1
007410	010246		MOV	%2, -(6)	:SAVE R2
007412	010346		MOV	%3, -(6)	:SAVE R3
007414	010446		MOV	%4, -(6)	:SAVE R4
007416	010546		MOV	%5, -(6)	:SAVE R5
007420	012704	0010C2	MOV	#LONUM,%4	:SET UP LONUM PCINTER
007424	012705	001012	MOV	#HINUM,%5	:SET UP HINUM POINTER
007430	011400		MOV	(4),%0	:SET R0 WITH LOW
007432	011501		MOV	(5),%1	:SET R1 WITH HIGH
007434	012703	177771	REPET4: MOV	#-7,%3	:SET SHIFT COUNT
007440	005002		CLR	%2	
007442	006300		SHIFT4: RSL	%0	:SHIFT PD LEFT AND
007444	006101		ROL	%1	:ROTATE CARRY INTO R1 AND
007446	006102		ROL	%2	:ROTATE CARRY INTO R2
007450	005203		INC	%3	:CHECK FOR DONE
007452	001373		BNE	SHIFT4	:CONTINUE SHIFT LOOP
007454	061402		ADD	(4),%2	:ADD NUMBER TO MAKE X 129
007456	005501		ADC	%1	:PROPOGATE CARRY
007460	061501		ADD	(5),%1	:ADD NUMBER TO MAKE X 129
007462	005502		ADC	%2	:PROPOGATE CARRY
007464	062700	001057	ADD	#1057,%0	:ADD LOW CONSTANT
007470	005501		ADC	%1	:PROPOGATE CARRY
007472	005502		ADC	%2	:PROPOGATE CARRY
007474	062701	047401	ADD	#47401,%1	:ADD HIGH CONSTANT
007500	005502		ADC	%2	:PROPOGATE CARRY
007502	062702	000006	ADD	#6,%2	:ADD HIGHEST CONSTANT
007506	060200		ADD	%2,%0	:REPRIME R0 WITH HIGHEST DIGIT
007510	005501		ADC	%1	:PROPOGATE CARRY
007512	010024		MOV	%0,(4)+	:SAVE R0
007514	010125		MOV	%1,(5)+	:SAVE R1
007516	020427	001012	CMP	%4,#LONUM+10	:CHECK FOR DONE ENOUGH
007522	001344		BNE	REPET4	:BRANCH IF NOT DONE
007524	012605		MOV	(6)+,%5	:RESTORE R5
007526	012604		MOV	(6)+,%4	:RESTORE R4
007530	012603		MOV	(6)+,%3	:RESTORE R3
007532	012602		MOV	(6)+,%2	:RESTORE R2
007534	012601		MOV	(6)+,%1	:RESTORE R1
007536	012600		MOV	(6)+,%0	:RESTORE R0
007540	000207		RTS	%7	:RETURN

007542	032767	002000	170020	.TRP:	BIT	#2000,SWP	
007550	001405				BEG	.ET	
007552	012767	000207	001416		MOV	#BELL,.TYPE	;TYPE A BELL
007560	000004	011176			TYPE,	.TYPE	
007564	005267	001414		.ET:	INC	ERRORS	;COUNT THE NUMBER OF ERRORS
007570	032767	020000	167772		BIT	#20000,SWP	;SKIP TYPEOUT IF SET
007576	001116				BNE	NHEAD	
007600	174267	171216			STF	AC2, ANS1	;SAVE THE ANSWER TO BE SURE
007604	000004	011102			TYPE,	RET	
007610	000004	011102			TYPE,	RET	
007614	011646				MOV	(6),-(6)	;PUT ADDRESS OF INSTRUCTION ON STACK
007616	162716	000002			SUB	#2,(6)	
007622	117667	000000	001356		MOVB	2(6),.WORDS	
007630	012605				MOV	(6)+,TTY	;TYPE (6)+ IN OCTAL
007632	004767	000546			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
007636	000004	011077			TYPE,	SPACE+3	
007642	012703	001002			MOV	#LONUM,%3	;SET UP POINTER
007646	004767	000174			JSR	7,STYPE	;TYPE A FLOATING POINT NUMBER
007652	000004	011105			TYPE,	\$SIGN	
007656	004767	000164			JSR	7,STYPE	;TYPE A FLOATING POINT NUMBER
007662	000004	011130			TYPE,	FPUAN	
007666	004767	000154			JSR	7,STYPE	;TYPE A FLOATING POINT NUMBER
007672	000004	011076			TYPE,	SPACE+2	
007676	016705	171140			MOV	FPS,TTY	;TYPE FPS IN OCTAL
007702	004767	000476			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
007706	105767	001274			TSTB	WORDS	;CHECK FOR STATUS ERROR
007712	00014				BPL	.STAT	;BRANCH IF NOT
007714	000004	011076			TYPE,	SPACE+2	
007720	016705	171120			MOV	FEC,TTY	;TYPE FEC IN OCTAL
007724	004767	000454			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
007730	000004	011076			TYPE,	SPACE+2	
007734	016705	171106			MOV	FPC,TTY	;TYPE FPC IN OCTAL
007740	004767	000440			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
007744	000004	011111		.STAT:	TYPE,	FORTAN	
007750	004767	000072			JSR	7,STYPE	;TYPE A FLOATING POINT NUMBER
007754	105767	001226		NHEAD:	TSTB	WORDS	
007760	001425				BEG	NHEAD	
007762	000004	011076			TYPE,	SPACE+2	
007766	016705	171056			MOV	\$FPS,TTY	;TYPE \$FPS IN OCTAL
007772	004767	000406			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
007776	105767	001204			TSTB	WORDS	
010002	100014				BPL	NHEAD	
010004	000004	011076			TYPE,	SPACE+2	
010010	016705	171036			MOV	\$FEC,TTY	;TYPE \$FEC IN OCTAL
010014	004767	000364			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
010020	000004	011076			TYPE,	SPACE+2	
010024	016705	171024			MOV	\$FPC,TTY	;TYPE \$FPC IN OCTAL
010030	004767	000350			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
010034	005737	177570		NHEAD:	TST	2\$SWR	
010040	100001				BPL	.+4	
010042	000000				HALT		
010044	000002				RTI		

```

010046 032767 001000 167514 $TYPE: BIT      #1000, SWR      ;CHECK TTY FORMAT
010054 001007      BNE      TYPE1
010056 105767 170766      TSTB     $FPS
010062 100432      BMI      TYPED
010064 004767 000076      JSR      7      TYPEF
010070 022323      TYPEA: CMP      (3)+, (3)+ ;UP DATE THE TYPEOUT POINTER
010072 000207      RTS      7

010074      TYPE1:
010074 012305      MOV      (3)+,TTY      ;TYPE (3)+ IN OCTAL
010076 004767 000302      JSR      %7,PRINTR ;TYPE LEADING ZERO'S
010102 000004 011100      TYPE,     SPACE+4
010106 012305      MOV      (3)+,TTY      ;TYPE (3)+ IN OCTAL
010110 004767 000270      JSR      %7,PRINTR ;TYPE LEADING ZERO'S
010114 105767 170730      TSTB     $FPS
010120 100363      BPL      TYPEA
010122 000004 011100      TYPE,     SPACE+4
010126 012305      MOV      (3)+,TTY      ;TYPE (3)+ IN OCTAL
010130 004767 000250      JSR      %7,PRINTR ;TYPE LEADING ZERO'S
010134 000004 011100      TYPE,     SPACE+4
010140 012305      MOV      (3)+,TTY      ;TYPE (3)+ IN OCTAL
010142 004767 000236      JSR      %7,PRINTR ;TYPE LEADING ZERO'S
010146 000207      RTS      7

010150 012346      TYPED: MOV      (3)+,-(6) ;GET WORD 1
010152 012346      MOV      (3)+,-(6) ;GET WORD 2
010154 012346      MOV      (3)+,-(6) ;GET WORD 3
010156 012346      MOV      (3)+,-(6) ;GET WORD 4
010160 012746 000022      MOV      #18,-(6) ;CHAR COUNT
010164 000406      BR      TYPE1
010166 012346      TYPEF: MOV      (3)+,-(6) ;GET WORD 1
010170 012346      MOV      (3)+,-(6) ;GET WORD 2
010172 005046      CLR      -(6) ;CLEAR WORD 3
010174 005046      CLR      -(6)
010176 012746 000010      MOV      #8,-(6) ;CHAR COUNT
010202 004767 000104      TYPE1: JSR      7,TY1 ;TYPE 1 BIT
010206 105766 000011      TSTB     9,(6) ;CHECK EXPONENT
010212 001001      BNE      .+4 ;BRANCH ON NON-ZERO EXPONENT
010214 005116      COM      (6) ;FLAG ZERO EXPONENT
010216 000004 011100      TYPE,     SPACE+4
010222 004767 000072      JSR      7,TY2 ;TYPE 2 BITS
010226 004767 000074      JSR      7,TY3 ;TYPE 3 BITS
010232 004767 000070      JSR      7,TY3 ;TYPE 3 BITS
010236 000004 011100      TYPE,     SPACE+4
010242 004767 000020      JSR      7,TYH ;TYPE 2 BITS
010246 004767 000054      TYPE2: JSR      7,TY3 ;TYPE 3 BITS
010252 005316      DEC      (6) ;DONE?
010254 001374      BNE      TYPE2
010256 022626      CMP      (6)+,(6)+ ;RESTORE
010260 022626      CMP      (6)+,(6)+ ;THE
010262 005726      TST      (6)+ ;STACK
010264 000207      RTS      7

```

010266	005766	000002	TYH:	TST	2(6)	:CHECK FOR ZERO EXPONENT FLAG
010272	100405			BMI	TYZ	:BRANCH ON ZERO EXPONENT
010274	012746	000001		MOV	#1,-(6)	:TYPE HIDDEN BIT AND ONE
010300	011667	000672		MOV	(6),TYPE	:FUDGE HIDDEN BIT
010304	000414			BR	TY+4	
010306	005166	000002	TYZ:	COM	2(6)	:GET RID OF ZERO EXPONENT FLAG
010312	012746	000001	TY1:	MOV	#1,-(6)	:TYPE 1 BIT
010316	000405			BR	TY	
010320	012746	000002	TY2:	MOV	#2,-(6)	:TYPE 2 BITS
010324	000402			BR	TY	
010326	012746	000003	TY3:	MOV	#3,-(6)	:TYPE 3 BITS
010332	005067	000640	TY:	CLR	.TYPE	
010336	006166	000006		ROL	6(6)	:SHIFT WORD 4
010342	006166	000010		ROL	8(6)	:SHIFT WORD 3
010346	006166	000012		ROL	10(6)	:SHIFT WORD 2
010352	006166	000014		ROL	12(6)	:SHIFT WORD 1
010356	006167	000614		ROL	.TYPE	:GET IT
010362	005316			DEC	(6)	:DONE?
010364	001364			BNE	TY+4	
010366	052767	000060 000602		BIS	#0,TYPE	:MAKE IT ASCII
010374	000004	011176		TYPE.	.TYPE	:TYPE IT
010400	005726		TS	(6)+		:RESTORE THE STACK
010402	000207		R'S	7		

STARTING POINT DIVIDE EXERCISES

Address	Hex	Assembly	Comment
000001	000130	PRINTR: MOVB	:SET ZERO FILL SWITCH
000002	000131	BR	
000003	000132	PRINTS: CLR	:SUPPRESS LEADING ZERO'S
000004	000133	MOV	:SET COUNT
000005	000134	MOV	:SAVE R4
000006	000135	MOV	:SET POINTER TO FIRST ASCII: CHARS
000007	000136	MOV	:CLEAR FIRST BYTE
000008	000137	MOV	:ROTATE FIRST BIT
000009	000138	MOV	:CLEAR BYTE OF CHARACTER
000010	000139	MOV	:ROTATE BIT INTO C
000011	000140	MOV	:PACK IT
000012	000141	MOV	:ROTATE BIT INTO C
000013	000142	MOV	:PACK IT
000014	000143	MOV	:ROTATE BIT INTO C
000015	000144	MOV	:PACK IT
000016	000145	MOV	
000017	000146	MOV	
000018	000147	MOV	
000019	000148	MOV	
000020	000149	MOV	
000021	000150	MOV	:CHECK FILL SWITCH
000022	000151	MOV	
000023	000152	MOV	:MAKE INTO ASCII: CHARS
000024	000153	MOV	:REPEAT
000025	000154	MOV	
000026	000155	MOV	
000027	000156	MOV	
000028	000157	MOV	
000029	000158	MOV	
000030	000159	MOV	
000031	000160	MOV	
000032	000161	MOV	
000033	000162	MOV	
000034	000163	MOV	
000035	000164	MOV	
000036	000165	MOV	
000037	000166	MOV	
000038	000167	MOV	
000039	000168	MOV	
000040	000169	MOV	
000041	000170	MOV	
000042	000171	MOV	
000043	000172	MOV	
000044	000173	MOV	
000045	000174	MOV	
000046	000175	MOV	
000047	000176	MOV	
000048	000177	MOV	
000049	000178	MOV	
000050	000179	MOV	
000051	000180	MOV	
000052	000181	MOV	
000053	000182	MOV	
000054	000183	MOV	
000055	000184	MOV	
000056	000185	MOV	
000057	000186	MOV	
000058	000187	MOV	
000059	000188	MOV	
000060	000189	MOV	
000061	000190	MOV	
000062	000191	MOV	
000063	000192	MOV	
000064	000193	MOV	
000065	000194	MOV	
000066	000195	MOV	
000067	000196	MOV	
000068	000197	MOV	
000069	000198	MOV	
000070	000199	MOV	
000071	000200	MOV	
000072	000201	MOV	
000073	000202	MOV	
000074	000203	MOV	
000075	000204	MOV	
000076	000205	MOV	
000077	000206	MOV	
000078	000207	MOV	
000079	000208	MOV	
000080	000209	MOV	
000081	000210	MOV	
000082	000211	MOV	
000083	000212	MOV	
000084	000213	MOV	
000085	000214	MOV	
000086	000215	MOV	
000087	000216	MOV	
000088	000217	MOV	
000089	000218	MOV	
000090	000219	MOV	
000091	000220	MOV	
000092	000221	MOV	

MACY11 270732 16-SEP-76 16:30 PAGE 47

44-NOV-11-00PL-3 FLOATING POINT DIVIDE EXERCISER
 00PL0.011 CROSS REFERENCE TABLE -- USER SYMBOLS

Year	1936	1937	1938	1940	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100																																																																																														
1936	1826	1828	1835	1840	1849	1850	1851	1852	1853	1854	1855	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2

MACY11 27(732) 16-SEP-76 16:30 PAGE 52

MANDEC-11-00FPL-2 FLOATING POINT DIVIDE EXERCISER
00FPL0.F11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

[illegible]

U-C FLOATING POINT DIVIDE EXERCISER
CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MACY11 27(732) 16-SEP-76 16:30 PAGE 53

JSR	468	469	541	542	616	617	691	692	768	769	346	847	928	929	1009
	1009	1092	1093	1167	1168	1244	1245	1324	1325	1412	1534	1546	1552	1552	1539
	1604	1725	1729	1730	1732	1735	1740	1743	1745	1750	1755	1758	1768	1774	1777
	1782	1785	1799	1804	1805	1806	1808	1809	1883						
LD	551	552	553	559	701	702	703	709	856	857	858	864	1018	1019	1020
	1026	1177	1178	1179	1185	1334	1335	1336	1342	1894	1896	1919	1921	1923	1924
	1925	1926													
LD	479	479	480	486	626	627	628	634	778	779	780	786	938	939	940
	946	1102	1103	1104	1110	1254	1255	1256	1262						
LDPS	477	481	550	554	625	629	700	704	777	781	855	859	937	941	1017
	1021	1101	1105	1176	1180	1253	1257	1333	1337	1927					
	1424														
MOV	430	431	433	436	439	440	441	442	443	444	445	446	447	448	449
	450	451	462	476	482	535	549	555	610	624	630	685	699	705	762
	776	782	840	854	860	922	936	942	1002	1016	1022	1086	1100	1106	1161
	1175	1181	1238	1252	1258	1318	1332	1338	1398	1406	1408	1410	1435	1436	1440
	1443	1448	1449	1450	1451	1454	1455	1456	1457	1460	1461	1464	1465	1466	1467
	1479	1482	1483	1484	1487	1488	1497	1555	1579	1609	1629	1630	1631	1632	1633
	1634	1635	1636	1637	1638	1639	1658	1659	1662	1663	1664	1665	1666	1667	1670
	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1699	1700	1703	1704	1705
	1706	1707	1708	1713	1721	1724	1727	1734	1739	1742	1749	1754	1757	1773	1776
	1781	1784	1788	1789	1790	1791	1792	1794	1795	1798	1819	1820	1823	1825	1827
	1845	1846	1869	1880	1886	1887	1898	1899	1900	1901	1902	1903	1904	1905	1908
	1912	1913	1914	1915	1916	1917	1929	1929	1936	1937	1944	1950	1951		
MOV	1423	1433	1480	1494	1723	1841	1844	1866	1940						
MOV	1413	1414	1415												
ROL	1642	1643	1683	1684	1829	1830	1831	1832	1833	1850	1852	1854			
ROLE	1851	1853	1855												
RTI	1407	1409	1417	1437	1444	1627	1762	1931	1952						
RTS	1462	1468	1503	1556	1581	1610	1668	1709	1770	1786	1815	1839	1870	1884	
SBC	519	592	593	594	669	744	745	746	824	904	905	906	986	1068	1069
	1070	1145	1220	1221	1222	1302	1384	1385	1386						
SETD	1889														
SET	566	590	598	716	742	750	871	895	910	1033	1059	1074	1192	1218	1226
	1349	1375	1390	1890	1891	1892	1893	1895	1897	1920	1922				
STF	493	517	523	641	667	673	793	817	928	953	979	990	1117	1143	1149
	1269	1295	1306	1718											
STEPS	490	563	638	713	790	868	950	1030	1112	1187	1264	1344	1616	1898	
STST	1121	1196	1273	1353	1617										
SUB	518	591	668	743	823	903	985	1067	1144	1219	1301	1383	1593	1592	1722
TRAP	390														
TST	432	488	561	636	711	788	866	948	1028	1113	1198	1265	1345	1441	1499
	1502	1623	1759	1814	1817	1838									
TSTB	1429	1485	1499	1736	1746	1751	1766	1778	1800	1856	1859	1938	1941		
	1931	1954	1955	1956	1957	1960									
.ASCIZ															
.ASECT	313														
.BLKW	1872														
.END	1977														
.EVEN	1931	1963													
.GLOBL	314														
.LIST	311	361	396	400	454	1395	1445	1628	1710	1763	1840	1895	1931	1935	
.MACR	361														
.MACRO	361														
.NLIST	311	361	396	400	454	1395	1445	1628	1710	1763	1840	1895	1931	1935	
.PAGE	1154	1259	1311	1395											
.REPT		296													

MO4

MAINDEC-11-DCFPUD-3 FLOATING POINT DIVIDE EXERCISER
DCFPUD.F11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MACY11 27(732) 16-SEP-76 16:30 PAGE 54

.SET 311 361 400 454 1395 1445 1628 1710 1763 1840 1885 1935
FILE 312

.ABS. 011212 000
000000 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*.DCFPUD.SEG/SOL/CRF/PAGNUM+DCFPUD
RUN-TIME: 6 12 2 SECONDS
RUN-TIME RATIO: 130/22=5.9
CORE USED: 9K (17 PAGES)

